

ABSTRACT

Title of Document: Sum-frequency Generation in Laser Safety and Quantum Telecommunications Applications

Jemellie Galang Houston, Master of Science, 2011

Directed By: Dr. Charles W. Clark

This thesis describes the implications of sum-frequency generation in both laser safety and quantum telecommunications applications. Green laser pointer technology uses frequency doubling of invisible 1064 nm infrared radiation to visible 532 nm green radiation. An inexpensive green laser pointer was found to emit infrared leakage primarily due to the lack of an infrared-blocking filter. An experimental setup using common household materials was presented to detect unwanted infrared radiation from such devices. Also reported, is the design and characterization of a high-speed versatile 780 nm pump source up to 1.25 GHz through second harmonic generation from a wavelength of 1560 nm. The 780 nm source is currently being used for the production of correlated photon pairs, one of which is at 656 nm, the hydrogen Balmer alpha line. The final goal will be to generate a high-speed entanglement source after some adjustments in the correlated pair source assembly. This will improve an operational quantum key distribution system.

SUM-FREQUENCY GENERATION IN LASER SAFETY AND QUANTUM
TELECOMMUNICATIONS APPLICATIONS

By

Jemellie Galang Houston

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2011

Advisory Committee:

Professor Michael Coplan, Chair
Adjunct Professor Charles W. Clark
Professor Luis Orozco

© Copyright by
Jemellie G. Houston
2011

Dedication

I would like to dedicate this to my mother, Carmelita Galang. She was my motivation. Without her, I would not have been where I am.

Acknowledgements

I would like to thank my advisor, Charles W. Clark, for all the words of encouragement and for all the opportunities he presented for me to explore the world beyond the walls of our laboratory. I want to extend my special thanks to Dr. Michael Coplan for getting me started on my path and helping me through my academic career. I also want to thank my colleagues Joshua Bienfang and Alessandro Restelli for guiding me through this whole ordeal every step of the way.

I must also thank JQI at the University of Maryland, together with the Chemical Physics Program and NIST for their sponsorship of my research and school fees and related expenses.

I would like to acknowledge the help of Danny Rogers, Caroline Martin, Michael Wayne, Alan Mink, Jay Fan, Jun Chen, and Alan Migdall. I am also thankful to Debbie Jenkins, Helen Felrice, and Christina Brown for all the administrative help they provided me during my stay.

I am grateful to Allan and Shelley Holt, for they are the first people to see my potential and give me the opportunity to harness it.

Lastly, I want to thank my husband Steven, the reason I was able to go through and overcome the hardest part of my life.

Table of Contents

Dedication.....	ii
Acknowledgements.....	iii
Table of Contents.....	iv
List of Figures.....	v
Chapter 1: Introduction.....	1
Fundamentals of sum-frequency generation.....	2
Second harmonic generation.....	3
Four-wave mixing.....	4
Chapter 2: Laser safety application.....	5
Why green laser pointers?.....	5
Principles of green laser pointer operation.....	6
Hazards of green laser pointers.....	8
Power measurements.....	10
Home-brewed experiment for detecting infrared leakage.....	15
Diffraction.....	16
Chapter 3: Quantum telecommunications application.....	18
What are the benefits?.....	18
Current operational quantum key distribution (QKD) system.....	19
Prospects for improvement.....	20
Compatible detectors.....	23
QKD in Vegas.....	24
QKD in Vegas II.....	24
Chapter 4: Versatile 780 nm pump source.....	26
How is it done?.....	26
Phase-matching optimization.....	28
Crystal alignment.....	31
Initial power measurements of the pulsed 1560 nm source.....	32
Maximizing the 780 nm peak power output.....	36
Chapter 5: High-speed generation of correlated photon pairs.....	39
What can we do with it?.....	39
Experimental layout of correlated pair generation.....	40
Coincidence measurements.....	42
Chapter 6: Conclusion.....	45
Future development.....	45
Lessons learned.....	45
Appendix A.....	47
Appendix B.....	54
Glossary.....	71
Bibliography.....	72

List of Figures

- Figure 1 Energy level transition diagram in a second harmonic generation process
- Figure 2 Energy level transition diagram in a four-wave mixing process where $\omega_s > \omega_p > \omega_i$
- Figure 3 Diagram of human visual response showing the green laser pointer line at 532 nm, and near the 650 nm wavelength of a typical red laser pointer.
- Figure 4 Atomic energy levels and light involved in the operation of a GLP
- Figure 5 Schematic of a the operation of GLP based on a multi-crystal assembly
- Figure 6 Power spectrum of the first laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.
- Figure 7 Power spectrum of the second laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.
- Figure 8 Power spectrum of the third laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.
- Figure 9a Power spectrum of the fourth laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.
- Figure 9b Power spectrum of the fourth laser under test, around the green wavelengths
- Figure 10 Experimental setup for determining whether infrared radiation is emitted by a GLP.
- Figure 11 Diffraction spots as observed from a regular camera insensitive to infrared and a modified webcam with no IR blocking filter.
- Figure 12 Color coded portrayal of the dark absorption lines in the solar spectrum from the National Optical Astronomy Observatory

- Figure 13a Normalized solar intensity as function of wavelength around H α line
- Figure 13b The seven transitions between the $n = 2$ and $n = 3$ states of atomic hydrogen that make up the H α feature
- Figure 14 Thin detector and thick detector detection efficiency vs. wavelength
- Figure 15 Experimental set-up for production of high-speed 780 nm pulses [MZM Mach Zehnder Modulator, EDFA Erbium Doped Fiber Amplifier, PPLN Periodically Poled Lithium Niobate, PBSC Polarization Beam Splitting Cube]
- Figure 16 Mathematical model for phase matching temperature using the Sellmeier equation and phase-matching condition relation.
- Figure 17 Spectral response of 1560 nm signal with varying temperature
- Figure 17b Measured 780 nm power with respect to varying PPLN oven temperature
- Figure 18 A comparison of peak power measurement from an attenuated input to the oscilloscope and calculated from average power using a tunable 1560 nm source
- Figure 19 Top: Spectral response from varying repetition rate to observe presence of amplified spontaneous emission (ASE) from the EDFA
Bottom: Spectral response around the maximum
- Figure 20 Spectral response of varying repetition rate without the presence of ASE
- Figure 21 A comparison of peak power measurement from an attenuated input to the oscilloscope and calculated from average power using a laser diode source
- Figure 22 1560 nm peak power measurement while maximizing 780 nm output with polarization control
- Figure 23 780 nm peak power measurement comparison with the unclear origin of polarization related effect
- Figure 24 Microstructure fiber structure with 2.3 μ m core diameter.
- Figure 25 Schematic of high-speed correlated pair generator

Figure 26 Nonlinear relationship between PCF output power and count rate in the 656.28 nm beam path suggesting four-wave mixing

Figure 27 Schematic of high-speed entangled pair generator

Chapter 1: Introduction

As our society adopts advanced technologies, a basic understanding of these technologies becomes necessary to protect ourselves from any inherent dangers they might impose. Two technologies developed over the past fifty years, lasers and inter-networked computing, have transformed our society, arguably for the better. However, they each present an inherent danger that much of society is ignorant of: a physical health risk with the use of some lasers and a virtual cybercrime risk with the use of inter-networked computing.

The invention of lasers has revolutionized several fields of research including those of medicine, military, and communications. It has changed the way we view and use light and has supported the development of nonlinear optics and the discovery of related phenomena such as sum-frequency generation. On a darker note, however, a 5 mW laser pointer can easily cause permanent eye injuries with prolonged direct exposure. Furthermore, laser pointers with higher power are easily becoming available in the market today for very affordable prices. For example, a 1W blue laser recently became available for purchase for only a few hundred US dollars [1]. The said advertisement portrayed it to be a mere toy that easily burns through balloons and plastic. Unmentioned was the fact that power at this magnitude can easily burn human skin in the path of the beam. In this thesis, we show that besides such obvious high-power hazards, improper control of the sum-frequency generation process within common green laser pointers can result in dangerous infrared leakage undetected by the human eye.

The development of inter-networked computing created the World Wide Web and the Internet, which as a result spawned today's Information Age. Today, with just a few clicks of the mouse, one can instantly communicate with a person on the other side of the globe, transfer money between bank accounts, and choose to purchase from a selection of products greater than that available at most local stores. With this explosion in Internet communications and financial transactions, a related explosion in cybercrime has occurred. Almost daily, sites such as slashdot.org detail a new cyber-attack that puts Internet communications and personal information at risk. To decrease this risk, cryptographic tools have been developed which make it difficult for computer criminals to access our private communications and personal information. Quantum key distribution (QKD) is a cryptographic tool that is provably secure using the laws of quantum mechanics. In this thesis, we show how sum-frequency generation can be used to produce correlated photon pairs, that can be an important component of some QKD protocols.

Fundamentals of sum-frequency generation

Sum-frequency generation is a phenomenon where that combines coherent laser optical input beams in a nonlinear medium to generate an optical output at a different wavelength from that of the incident optical beams. Conservation of energy in this process implies that

$$\hbar\omega_3 = \hbar\omega_1 + \hbar\omega_2 \quad (1.1)$$

where ω_1 and ω_2 are the input pump field frequencies. The process is often used in green laser technology as well as entanglement generation. It requires a medium that responds nonlinearly to optical interactions, such as potassium titanyl phosphate (KTP) and periodically poled lithium niobate (PPLN) crystals. Additionally, a certain phase-matching condition between the interacting electromagnetic waves along the direction of propagation must be satisfied. For the process to be efficient, the phase mismatch between the fundamental frequency and its sub harmonics must be near zero. This phase mismatch is derived from the conservation of momentum and angular momentum condition for any non-linear process.

Second harmonic generation

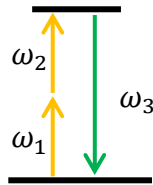


Figure 1. Energy level transition diagram in a second harmonic generation process

Second harmonic generation (SHG) is a special case of sum-frequency generation is when the new light beam produced is half the wavelength and thus twice the energy of the input monochromatic light beam. The process is mediated by an intrinsic property of the optical medium: the second-order nonlinear susceptibility, $\chi^{(2)}$. Thus, $\omega_2 = \omega_1 = \omega_3/2$ in equation 1.1.

Four-wave mixing

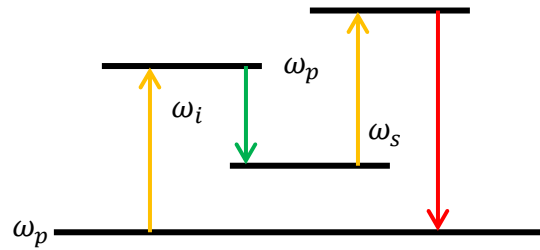


Figure 2. Energy level transition diagram in a four-wave mixing process where $\omega_s > \omega_p > \omega_i$

Four-wave mixing (FWM) is another special case of sum-frequency generation, four-wave mixing, allows for the production of a fourth coherent field from the interaction of three coherent laser optical beams. The third-order nonlinear susceptibility, $\chi^{(3)}$, governs a process whereby two photons at different wavelengths, called the signal and the idler, can be created simultaneously in a nonlinear medium from two photons from the input field. The energy relationship between the four interacting fields is shown in Figure 2.

Chapter 2: Laser safety application

Why green laser pointers?

Green laser pointer (GLP) technology is a common real-world application of second harmonic generation. These pointers are readily available in an inexpensive cigar-sized package to any consumer in the U.S market. They are typically used in dark places such as a large auditorium or an open field for nighttime star gazing because the human visual eye response peaks near 550 nm. The blue curve in Figure 3 describes the model of the photopic human luminosity, a measure of the perceived brightness of light to the human eye from a source of uniform intensity across the wavelength range. It indicates a luminosity coefficient of 0.8832 at 532 nm, and 0.107 at 650 nm [2]. This means that a 5 mW green laser pointer would give the same perceived brightness to the human eye as a 41 mW red laser pointer.

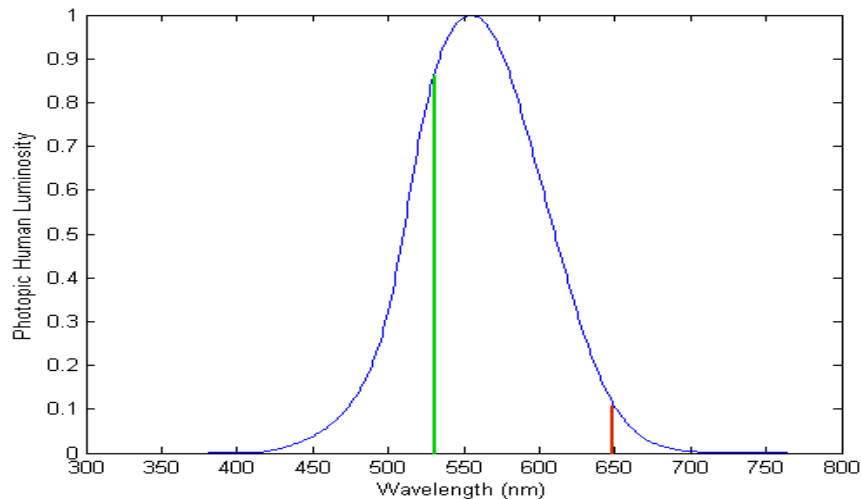


Figure 3. Diagram of human visual response showing the wavelength of the GLP line at 532 nm, and the 650 nm wavelength of a typical red laser pointer [2]

A colleague purchased three green laser pointers with the same power ratings to be given away as Christmas presents, one of which appeared dimmer than the others. To identify the reason for this irregularity, we measured the power emitted by the device. In our investigation, we found the inexpensive green laser pointer emitting ten times more invisible infrared radiation than visible green light. This is dangerous because visible radiation activates the blink reflex while infrared radiation remains undetected until potentially serious eye damage has been incurred. Our findings were summarized in *Technical Note 1668* of the National Institute of Standards and Technology [3]. This publication reported the first quantitative measurements of the infrared leakage problem and attracted wide attention, including reports in *WIRED Magazine*, *Optics and Photonics News* [4], SPIE Newsroom [5], msnbc.com and numerous internet sites. Following advice from colleagues at the U.S. Food and Drug Administration, the regulatory agency responsible for laser safety, we presented our results at the 2011 International Laser Safety conference. [6]

Principles of green laser pointer operation

The three elements essential to green laser technology are: a semiconductor pump laser at 808 nm; a neodymium-ion oscillator that emits 1064 nm radiation; and a nonlinear frequency-doubling crystal that produces 532 nm light. These three components are integrated into a small package with other optical elements, and the package is manufactured in large volume.

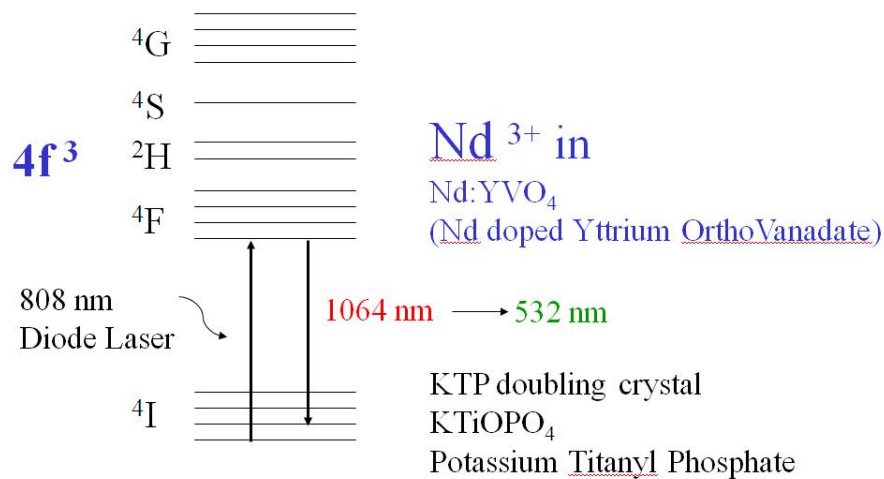


Figure 4. Atomic energy levels and light involved in the operation of a GLP. Figure credit by Joseph Reader, NIST

Triply-charged ions of the neodymium atoms, Nd^{3+} , are present as dopants in a crystal of yttrium orthovanadate (Nd:YVO_4). As diagramed in Figure 4, the Nd^{3+} ion contains three $4f$ electrons outside closed electron shells, so its electronic configuration is designated $4f^3$. The horizontal lines indicate some of the energy levels of the $4f^3$ configuration, labeled by conventional spectroscopic notation. A diode pump laser with an infrared wavelength of 808 nm excites the lowest $4I$ state to an electronically excited $4F$ state. The Nd^{3+} ion emits infrared radiation, at a wavelength of 1064 nm, due to the stimulated emission from the excited state into a different $4I$ state. This radiation is directed into a “frequency doubling” crystal of potassium titanyl phosphate, which uses SHG to emit light at half the wavelength: 532 nm.

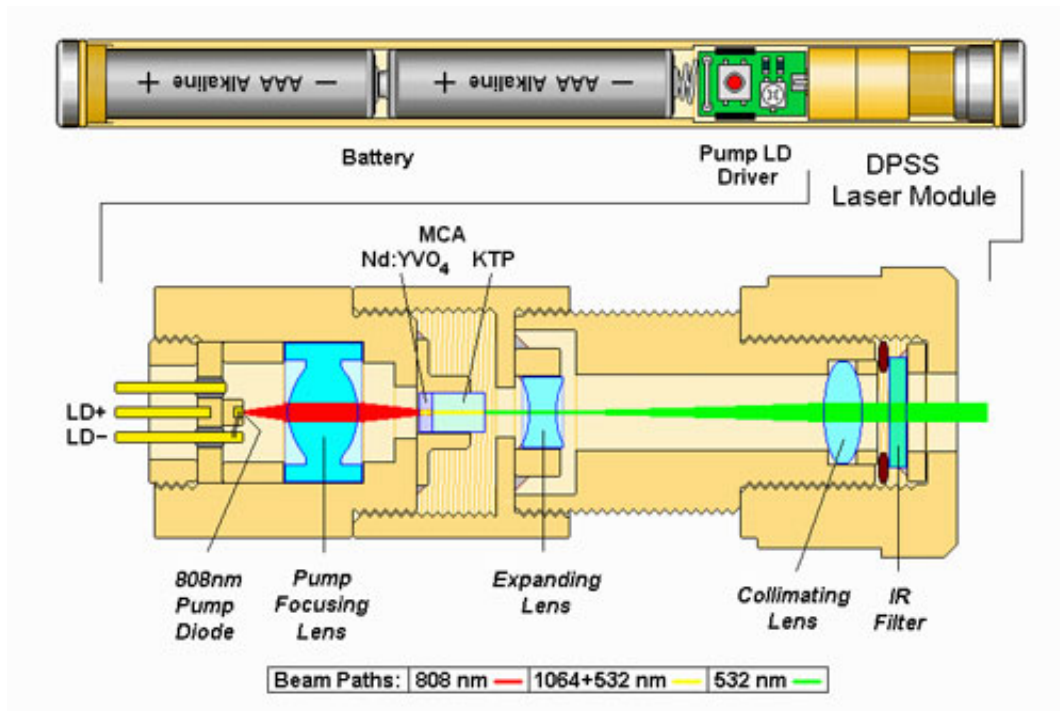


Figure 5. Schematic of a the operation of GLP based on a multi-crystal assembly
 Figure credit by Sam Goldwasser[7]

The Diode Pumped Solid State (DPSS) laser module shown in Figure 5 is an example of a safe GLP design. The 808 nm pump laser diode is optically coupled via a pump focusing lens to the Nd:YVO₄ conversion crystal, which is depicted as a violet component of the multiple-crystal assembly (MCA). The conversion crystal emits 1064 nm light into the KTP frequency-doubling crystal (light blue section). The 532 nm light from the KTP crystal is sent through both expanding and collimating lens assemblies to produce a collimated output beam. In this configuration, an IR filter prevents the 808 nm and 1064 nm light from exiting the laser.

Hazards of green laser pointers

A 10 mW laser, rated class IIIb [8], is capable of causing severe eye injuries regardless of its wavelength. As mentioned earlier, this is typically avoided by

following the beam path and directing it away from the eyes. Moreover, the human blink reflex system acts as a defense for protection against visible radiation. However, the blink reflex will not be activated by infrared radiation since the human eye does not perceive infrared wavelengths (shown before in Figure 3). Thus, a GLP emitting a significant portion of its radiation at the infrared wavelengths can cause serious eye injuries if reflected into a bystander's eyes.

The hidden danger associated with GLPs occurs in cases of low conversion efficiency from infrared to visible radiation. This can result from the manufacturing process. Normally, if the conversion efficiency is high, the inter-cavity infrared power will be low because conversion to green light draws down the power of the infrared. However if the conversion efficiency is low, then the intra-cavity 1064 nm power can build up to high levels, resulting in strong infrared emission. In the extreme case of zero conversion efficiency, it would be possible for the GLP to emit intense infrared light but no visible green light. Inclusion of an infrared-blocking filter in GLP design can prevent such infrared emission as given in Figure 5. However, upon disassembly of the laser under test, we found no such filter integrated in the design. Additionally, We did not even find a holder for such an important optical element. Accesses to tutorials to increase the power of the green laser pointer are easily available in the internet (i.e. youtube.com) by increasing the pump diode intensity. This makes these particular laser pointers even more hazardous.

One particular hazard associated with such an output is due to the fact that modern buildings have special window coatings to reflect infrared radiation from the sun. This material would cause the infrared part of the beam to be reflected and green

light transmitted through the glass. Any differential reflection of infrared vs. green light poses such hazards, since the green light that travels with the infrared light may not be sufficiently intense to activate the blink reflex. Invisible lasers, in the infrared and ultraviolet, are among the most frequently reported sources of laser-induced eye injuries, with many reported instances of permanent damage to vision [9]. A survey of the 100 accidental, non-medical laser-induced eye injuries that had been reported in the scientific literature up to 1999 found that Nd-based lasers with wavelengths of 1060-1064 nm were implicated in 49 % of all cases, while 532 nm lasers accounted for only 7 %. [10]

Power measurements

These measurements were done using a thermal detector that converts the heat absorbed from the laser beam per unit time to a power quantity. The inherent uncertainty due measurement accuracy of the device is $\pm 0.5\text{mW}$. The total emitted power of the laser under test, measured by the thermal detector, was 20 mW. From a separate measurement with a spectrum analyzer, we find that the combined power at invisible infrared composed of 808 nm pump and 1064 nm, is more than ten times the power at the visible 532 nm green radiation, as shown in Figure 6. This means that, for the laser pointer under test, only 1.5 mW is green and 18.5 mW is emitted at infrared wavelengths. Infrared radiation at such intensities is extremely dangerous. Other uncertainties associated with the following plots are due to the 98% coupling efficiency of green laser pointer into the fiber.

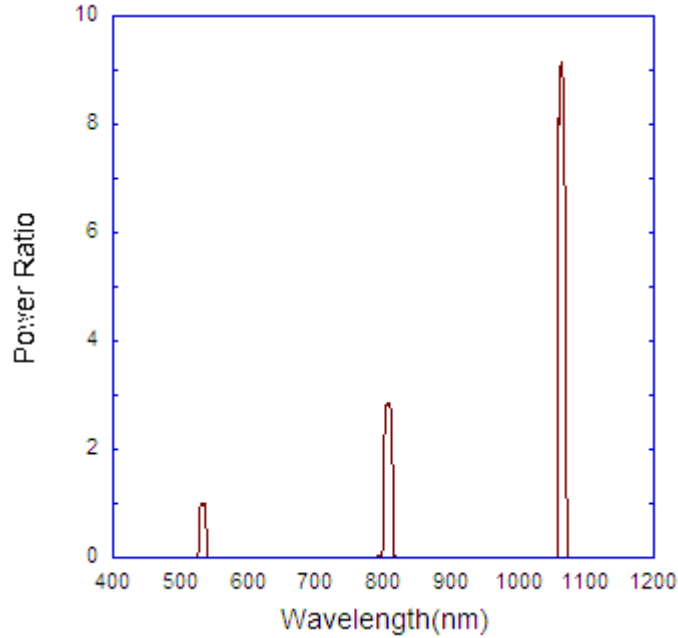


Figure 6. Power spectrum of the first laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.

In the interest of knowing the implication of this discovery, we ordered several other green laser pointers of the same type and performed the same power measurements. The results were highly variable. In one laser pointer, multiple peaks were observed around the green radiation. Another laser pointer has the same infrared radiation problem, though with better power ratio between the 1064 nm and 532 nm without the 808 nm contribution. I can only speculate that alignment optimization is not a priority in the manufacturing of these devices due to the inconsistencies between each power spectra obtained from several green laser pointers.

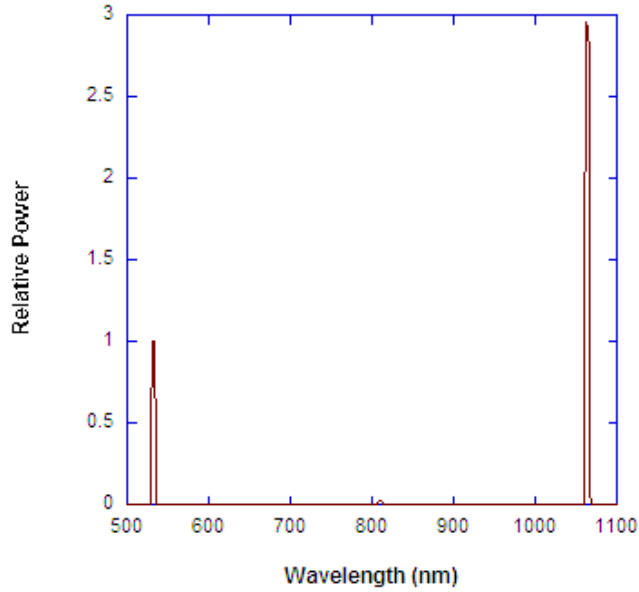


Figure 7. Power spectrum the second laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.

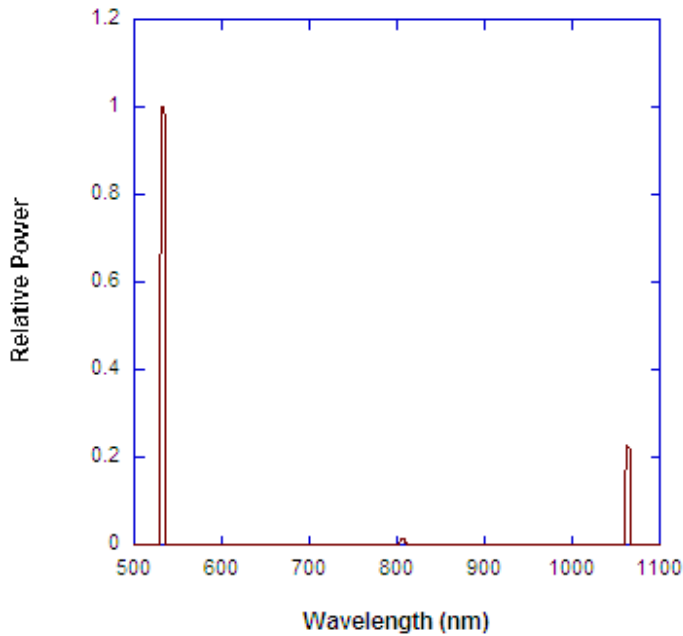


Figure 8 Power spectrum the third laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.

The second laser pointer under test showed a 3:1 ratio of the 1064 nm to 532 nm radiation as illustrated in Figure 75. The total thermal power emitted by this laser

pointer was 10 mW. This means that about 2.5 mW was in the visible spectrum and 7.5 mW was in the infrared. The third laser pointer we tested had more green with a 1:5 ratio of 1064 nm to 532 nm seen in Figure 8. Apparently, it appears to be an acceptable laser pointer. However, the total power measured using the thermal detector was 20 mW. That is still about 4 mW of infrared emission.

An unexpected sum-frequency generation was found in the fourth laser pointer we tested. The total thermal power emitted by this laser pointer was also 20 mW. The observed several significant peaks around the 1064 nm wavelength, as illustrated in Figure 9a, can be attributed to two different species of Nd caused by an imperfection in the crystal. The two peaks correspond to 1064 nm and 1084 nm, the previous is a known transition for a Nd:YVO₄ while the latter is a known transition for a Nd:LiNO₃ [11]. Upon closer look around the green wavelengths, we observe three significant peaks as shown in Figure 9b. These peaks were measured to be at 532 nm, 536 nm, and 542 nm wavelengths. The 532 nm and 542 nm peaks are due to the frequency doubling of 1064 nm and 1084 nm respectively. The 536 nm peak is due to the sum-frequency contribution of both 1064 nm and 1084 nm. The multiple peaks can also be attributed to poor phase matching, an effect that will be described in more detail in chapter 4.

In all green laser pointers we tested, the presence of unwanted 1064 nm poses substantial hazard despite the varying power output. Power ratings on all green laser pointers tested besides the ones with provided power spectrum in this section, were >20 mW. The measured power ratings varied from 10 mW to 30 mW, which violates safety standards that require explicit warning of power outputs from such devices.

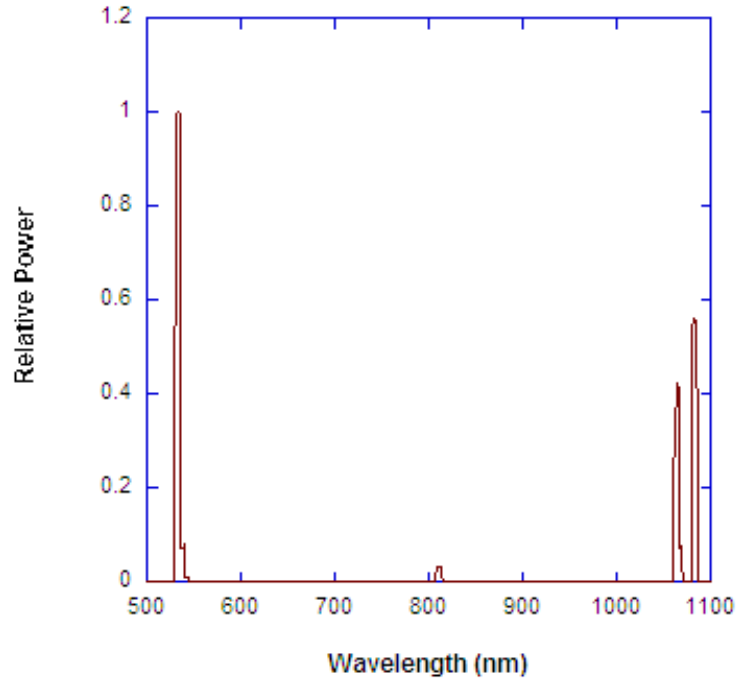


Figure 9a Power spectrum of the fourth laser under test, showing the ratios of the measured power of each laser line to the measured power of the green line at 532 nm.

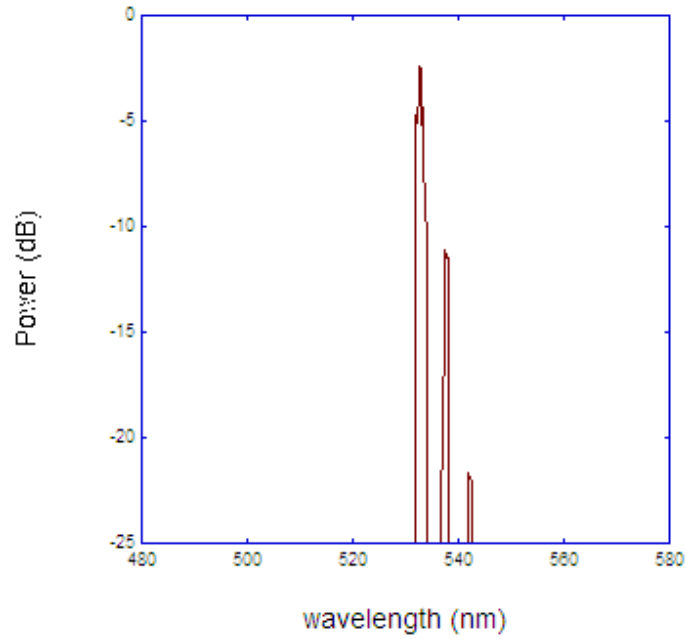


Figure 9b Power spectrum of the fourth laser pointer under test near the 532 nm wavelength

Home-brewed experiment for detecting infrared leakage

We now describe a simple test to observe infrared leakage using household items. This test can be performed by anyone who has access to a portable web camera, TV remote control, regular camera, and a CD. The CD was used as a diffraction grating to deflect light of different wavelengths in different directions [12]. Most of the CD surface was covered with black tape to avoid unwanted reflection to uncontrolled directions. A modified webcam was used as infrared detectors to locate unwanted infrared radiation. An infrared TV remote control unit was used to determine whether a webcam is infrared sensitive. If it is not, instructions to remove the infrared filter are easily available on internet search engines and video tutorials are also accessible. Cable ties were used to keep the laser ON during of the measurements and experiment. Plastic cups and a stack of books were used as optical mounts for the laser pointer, CD and camera.

A photograph of the experimental setup using common household items is presented in Figure 10. The GLP light passes through a hole in the paper screen and is diffracted by the black-masked CD on the left. Five green diffraction spots were visible on the paper screen. The diffraction pattern generated is photographed by a digital camera usually sensitive only to visible light and a modified webcam sensitive both visible and near infrared light.

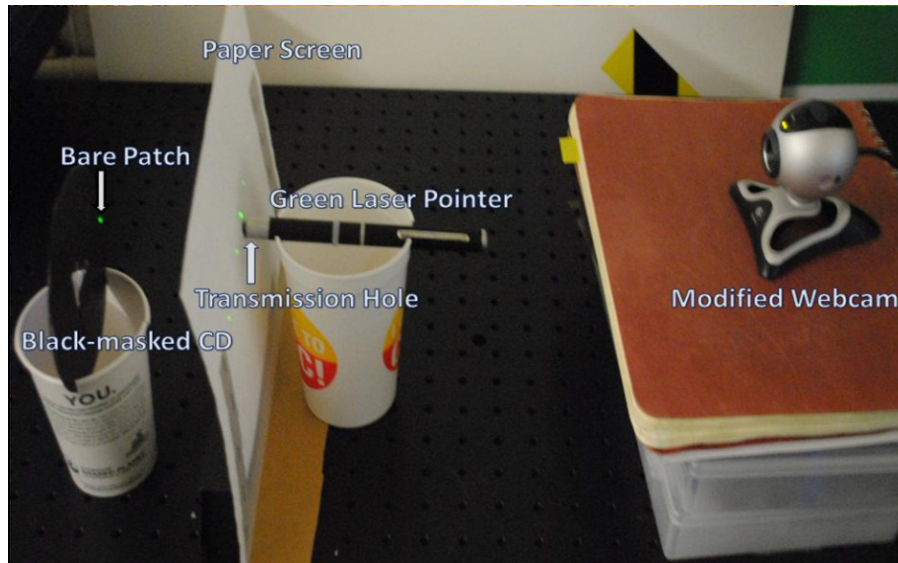


Figure 10. Experimental setup for determining whether infrared radiation is emitted by a GLP.

Diffraction

The CD acts as a diffraction grating with a spacing defined by the standard CD track separation of $1.6 \mu\text{m}$ [6]. It is possible to calculate the angle of diffraction Θ of reflected light as a function of its wavelength λ and the grating spacing d . For normal incidence light, the simple relationship is

$$\sin \theta = m\lambda/d \quad (3.1)$$

[13] where $m = 0, \pm 1, \pm 2$, etc. is an integer. The formula indicates that the incident light will be reflected back in a discrete series of angles corresponding to the integral values of m , where the case $m = 0$ corresponds to simple back reflection. The Figure 11 contrasts the visible and invisible spectrum of the reflected light.

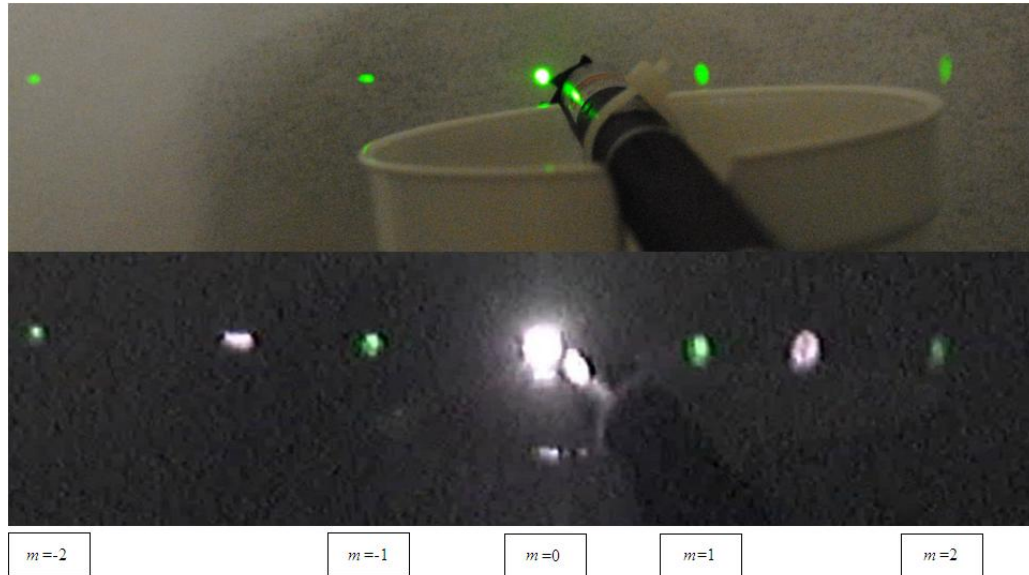


Figure 11. Diffraction spots as observed from a regular camera insensitive to infrared and a modified webcam with no IR blocking filter.

The top frame of Figure 11 shows the photograph of the diffraction using a cellphone camera that is insensitive to infrared radiation. This corresponds closely to what would be seen by the naked eye. The bottom frame is the photograph of the same diffraction from the viewpoint of the IR sensitive camera, modified webcam by removing its IR-blocking filter. Each spot is labeled by its corresponding order of diffraction. As expected, only the green light is seen, distributed left to right among diffraction orders of $m = -2, -1, 0, 1$ and 2 . Now we can see reflected 808 nm infrared light, with a central ($m = 0$) reflection much more intense than the green light's central reflection, and with bright $m = \pm 1$ reflections outside the fainter $m = \pm 1$ reflections of the green light. The webcam was not sensitive to wavelengths of 1064 nm, so we cannot see the diffraction spots ($m = \pm 1$) corresponding to this wavelength. Furthermore, if detectable, such spots would overlap the $m = \pm 2$ spots of the 532 nm green light.

Chapter 3: Quantum telecommunications application

What are the benefits?

In a typical example of a symmetric encryption, a message (the plaintext) is transformed (encrypted) into a new message (the ciphertext) using a string of random bits (the secret key). A crucial property of the transformation is that it should be difficult to back-transform (decrypt) the ciphertext back into the original plaintext without knowledge of the secret key. An example of a provably secure symmetric encryption scheme, given that the bits of the secret key are completely random and never reused, is the one-time pad, which XORs the plaintext and the secret key to create the ciphertext. A problem with any symmetric encryption scheme, including the one-time pad scheme, is how to confidentially distribute the secret key the two remote communication parties without a third party eavesdropping. Additionally, as with the one-time pad, the security of any symmetric encryption scheme is no better than the randomness of the secret key bits. Quantum key distribution (QKD) addresses the first problem by providing a provably secure method of establishing a secret key between two parties, in which detection of an eavesdropper is guaranteed by quantum physics. The QKD protocol described below (BB84) was first introduced by Bennett *et al.* [14].

Current operational quantum key distribution (QKD) system

NIST currently has an operational QKD system that uses an attenuated 850 nm single photon source [15]. The transmitter is referred to as Alice and the receiver as Bob. They are both made of various electronic and optical systems necessary to implement the QKD protocol. They are connected by a quantum channel with a unidirectional free space quantum channel from Alice to Bob and a classical channel using Ethernet and fiber. Alice and Bob also have optical elements that can send and measure, respectively, photons in one of two polarization bases: linear or circular. In linear basis, photons can be horizontally $|H\rangle$ or vertically $|V\rangle$ polarized, and in the circular basis, photons can be clockwise $|CW\rangle$ or counter-clockwise $|CCW\rangle$ polarized. For each basis, a bit value of 0 and 1 is consistently assigned to its possible polarizations. Using a random number generator, Alice selects both a basis and its polarization with which to send a photon in the quantum channel. Independently, Bob randomly chooses a basis to measure the arriving photon. They exchange information regarding the basis they sent and measured the photon through the open classical channel. If Bob measured it in a different basis from which Alice sent it, they disregard the bit and repeat the same procedure. If they measured in the same basis, the bit is added to the secret key. The process mentioned is called sifting.

As mentioned earlier, it is possible to discover the presence of an eavesdropper, whom we will call Eve, during a communication between Alice and

Bob. An interception of a message by Eve is, quantum mechanically, a measurement of the polarization of the photon, which changes the photon's polarization to that of the pass axis of the polarization analyzer. Eve must then send another photon to Bob without knowledge of the polarization state that Alice originally transmitted. Hence, there will be a 50% probability that the new photon is in the same basis and a 25% probability that it is also in the correct polarization state that Alice sent. When error correction and privacy amplification processes are performed, all errors beyond a threshold are attributed to Eve (details of these processes are beyond the scope of this thesis but can be found in [16]). It is then possible to manually stop the communication between Alice and Bob if the error rate is too high and the presence of Eve is detected.

Prospects for improvement

The current QKD system has enough throughput to do a broadband one-time-pad encryption of a video at a rate of 1 Mb/s during the nighttime and 100 Kb/s during the daytime. The performance degradation during daytime is due to undesired noise from photon emissions from the sun. Within the sun's optical spectrum, dark absorption lines known as Fraunhofer lines exist, as shown in Figure 12; among the over five hundred Fraunhofer lines, the hydrogen H_α feature at 656.28 nm is particularly promising for free space transmission (this is also known as the Balmer α feature). Figure 13a shows that H_α produces an 80% notch in the solar spectrum over a wavelength range of 0.12 nm (or 1.2 Ångstrom units, as labeled there).

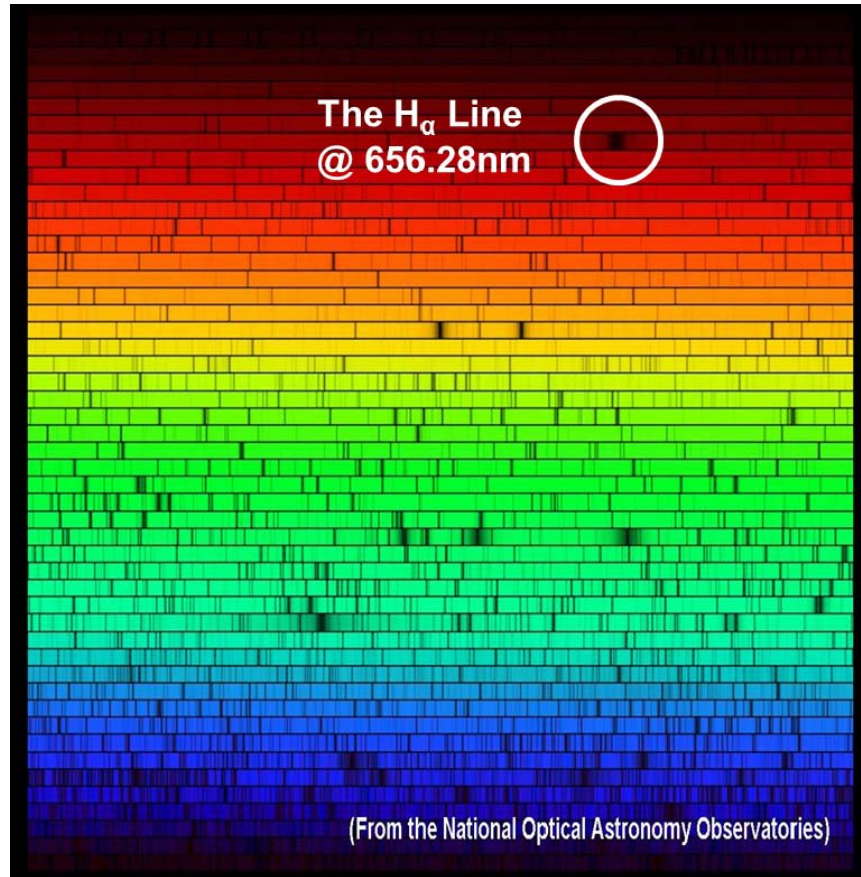


Figure 12. Color coded portrayal of the dark absorption lines in the solar spectrum from the National Optical Astronomy Observatory[17]

The H_{α} feature is actually an array of seven absorption lines associated with transitions between the $n = 2$ and $n = 3$ states of atomic hydrogen as shown in Figure 13b. The $2s_{1/2}$ and $2p_{1/2}$ are expected to be degenerate but quantum electrodynamics proved a tiny energy splitting called lamb shift due to the fluctuations of the electromagnetic field in the vacuum causing a perturbation in the position of the electron. The 1.2 angstrom notch shown in Figure 13a can be viewed as a relatively noiseless channel for quantum communications. However, a laser source at the H_{α} line, at GHz rate and entangled, is not a readily available product in the market. Production of photons at this wavelength may be done in the laboratory using

nonlinear processes such as parametric down conversion and four-wave mixing. A methodology for this production process will be discussed in more detail in Chapter 5.

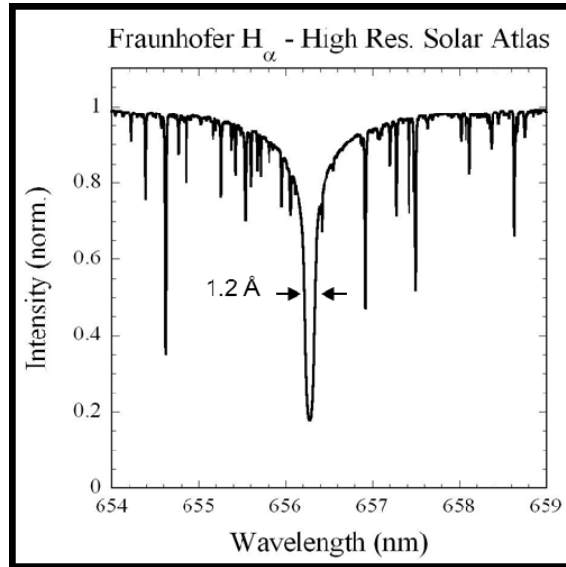


Figure 13a. Normalized solar intensity as function of wavelength around H_{α} line [18]

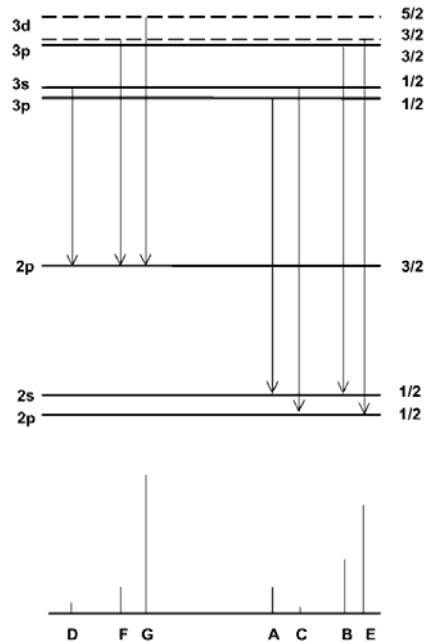


Figure 13b. The seven transitions between the $n = 2$ and $n = 3$ states of atomic hydrogen that make up the H_{α} feature [19]

Compatible detectors

It is equally important to have appropriate detectors for an overall efficient QKD system. Working at different wavelengths demands the use of different detectors, as indicated in Figure 14[20][21]. Thin single photon avalanche diode (SPAD) has reduced efficiency in the near infrared (NIR) regime, which makes it less favorable in working with the hydrogen Balmer alpha. However, 30% to 35% detection efficiency for the thin SPADs can still be acceptable. Using a Perkin Elmer (Thick) detector can increase the detection efficiency from about 40% to 60% while looking at the H_{α} line. However, thin Si SPADs are more economical and robust, which is why they are still commonly used. The thick and thin nomenclature refers to the depth of the depletion region. The thicker detector has higher timing jitter, which is a temporal uncertainty. Thin detectors have better timing resolution which allows their operation at higher transmission rates.

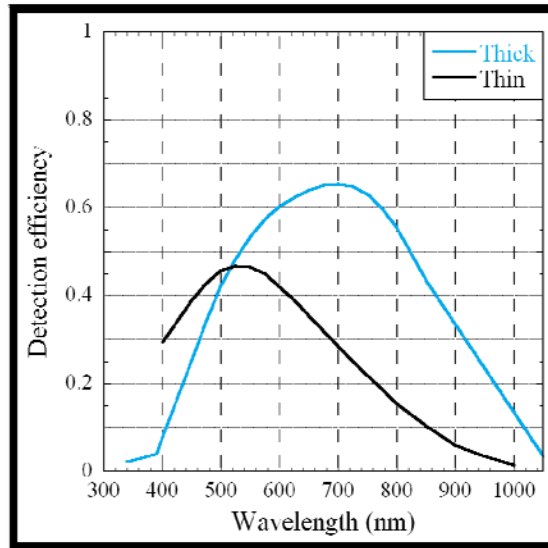


Figure 14. Thin detector and thick detector detection efficiency vs. wavelength[20][21]

QKD in Vegas

Since QKD enables what is arguably the most secure type of communication, the system is well-accepted in the dark side of the cyber community: computer hackers. We did a live demonstration of the free space quantum key distribution inside a hotel in Las Vegas, Nevada for the Black Hat (Caesar's Palace) and DEFCON 16 (Riviera Hotel) conferences. We took Alice and Bob (the physical transmitting and receiving subsystems of the QKD link) and three pallets of their equipment for this adventure to stream a live video feed using encrypted keys distributed via QKD. I made a graphical user interface that interacted with Bob's electronics through TCP/IP to extract sifted/secret key rates as well as error rates for visualization and diagnostic use during system operation (see Appendix A). We used polarizers to simulate eavesdropping events because they alter the initial polarization state prepared by Alice. We then observed from the user interface the errors caused by the polarizers' eavesdropping. The audience was awed, particularly when the error rate surpassed the threshold. This caused the system to stop generating new bits and end the live video feed.

QKD in Vegas II

The following year, we went back to the DEFCON 17 conference to further educate our colleagues in the computer science arena. We emphasized the importance of the random numbers in QKD by comparing a true random number generator (TRNG) with a pseudo random number generator (PRNG). PRNGs are made of mathematical algorithms which create a long series of seemingly random numbers from an initial designated number called a seed. On the other hand, TRNGs

use the inherent randomness in underlying devices or physical systems to generate random numbers.

For the demonstration, I created a graphical user interface that calculates the value of pi using a Monte Carlo simulation. It is a statistical approach that randomly puts points within a square of side length L , which has an inscribed circle of radius $L/2$. The ratio of the number of points inside the circle to inside the square should approximate $\pi/4$. In this process, a TRNG produces a more exact value of pi as compared to the result produced by a PRNG. I also created a graphical user interface that showed both the repeatability and predictability of the PRNG output (see Appendix B). The string of what seems to be random numbers may be reproduced as long as the PRNG seed is determined. A QKD system is therefore vulnerable to attack when a PRNG is used for the selection of the polarization state to send or measure the photon. Unfortunately, the TRNG we used for the demonstration was capable of producing high entropy bits at 100MHz, while our QKD system needs bits produced at 1.25 GHz. Thus, exploring the possibilities of higher speed TRNGs is an important area for future research.

Chapter 4: Versatile 780 nm pump source

How is it done?

In order to achieve an entanglement source at 656.28 nm, we must first develop a pulsed 780 nm source with good performance and versatility for reasons to be explained in chapter 5. Second harmonic generation from a broadband telecommunication wavelength source at 1560 nm will produce 780 nm coherent radiation. Our proposed system has variable pulse width, with a minimum 45 ps, and repetition rate varying from 19 MHz to 1.25 GHz. These characteristics will be suitable with the current operational QKD system. The short pulse will complement the detector's timing resolution and the high repetition rate capability is also suitable to the current high-speed QKD system. Shorter pulses are desired to match the strong temporal gating (small widths for detection events) of the detectors in use. The schematic for the pulsed 780 nm production is demonstrated in Figure 15 [22].

Optical pulses are carved from the modulation of the 1560 nm continuous wave (CW) laser using a Mach Zehnder amplitude modulator (MZM) with a 20 GHz bandwidth. The modulator is driven by a custom pulse board that works as follows: Using an external synchronized clock, a square-wave signal and its delayed inverse are sent to a 13 GHz AND logic gate to produce variable pulse width; the pulse duration can be varied by varying the delay between the square wave and its inverse, which consequently reduces the overlap at the logic gate inputs. It is capable of producing pulses at repetition rates from 19 MHz to 1.25 GHz using a clock divider

circuit and 45 ps minimum pulse duration at the full width half maximum (FWHM). The modulated optical signal is split into a 99% component and a 1% component for analysis and diagnostics. The 99% output is amplified by a 1 watt average erbium doped fiber amplifier (EDFA) to have more 1560 nm power available for the 780 nm generation. Meanwhile, the 1% component is either used as the input to an oscilloscope or an optical spectrum analyzer. An optical isolator (OI) is used to prevent unwanted back reflection in the EDFA. Polarization paddles are used to control the polarization of the light in and out of the EDFA. There is a polarization paddle before the EDFA because the input polarization into the EDFA affects the output power [23]. On the other hand, the polarization paddle after the EDFA is used to correct any polarization changes due to the fiber inside the EDFA.

Collimated light from the output fiber is passed through a polarization beam splitting cube (PBSC) and a half wave plate to choose the extraordinary axis of propagation in the PPLN, This axis has a higher conversion efficiency than the ordinary axis in the conversion crystal used. A lens is placed at the input and output of the crystal, referred to as the focusing and collecting lens, respectively, to locate the point of maximum buildup in the crystal. At the output, a dichroic mirror is used to separate the 1560 nm and 780 nm light by transmitting and reflecting these components, respectively. A band pass filter is placed after the dichroic to remove any reflected 1560 nm light and have a purer 780 nm signal.

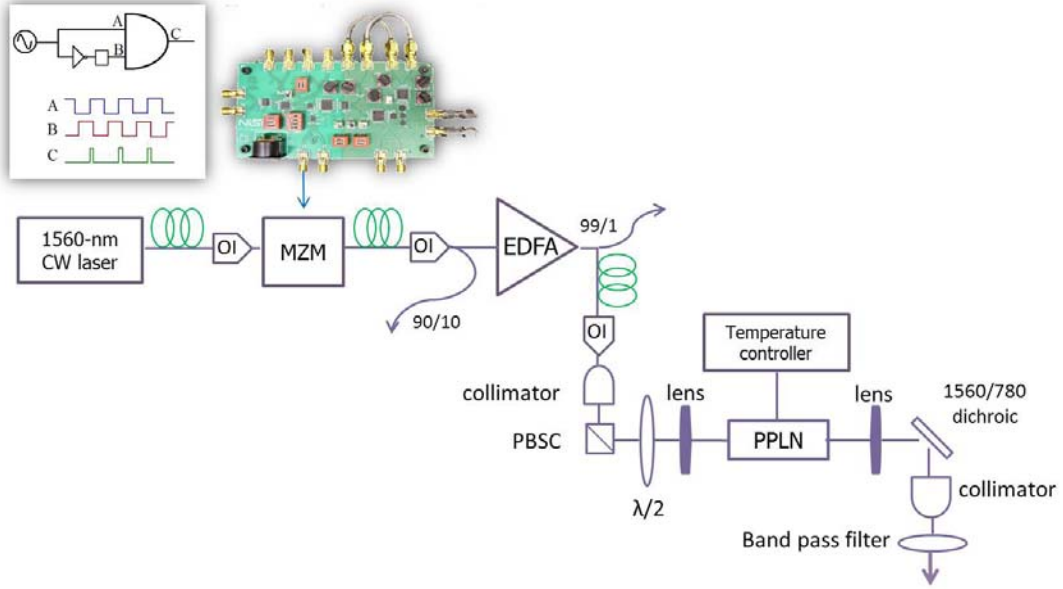


Figure 15. Experimental set-up for production of high-speed 780 nm pulses [MZM Mach Zehnder Modulator, EDFA Erbium Doped Fiber Amplifier, PPLN Periodically Poled Lithium Niobate, PBSC Polarization Beam Splitting Cube]

Phase-matching optimization

Since the SHG process involves mixing several electromagnetic waves in a nonlinear crystal, a phase matching condition must be satisfied to resolve phase difference between the distinct frequencies present in the medium. This condition is satisfied when $\Delta k_{opt} = 0$ in Equation 5.1 below, where Λ is the polling period

$$2\pi \left(\frac{n_e(\lambda_1, T)}{\lambda_1} - \frac{n_e(\lambda_2, T)}{\lambda_2} - \frac{n_e(\lambda_3, T)}{\lambda_3} - \frac{1}{\Lambda(T)} \right) = \Delta k_{opt} \quad (5.1)$$

The index of refraction n_e inside the conversion crystal varies with temperature and wavelength and can be found using the Sellmeier equation,

$$n_e^2 = a_1 + b_1 f + \frac{a_2 + b_2 f}{\lambda^2 - (a_3 + a_4 f)^2} + \frac{a_4 + b_4 f}{\lambda^2 - a_5^2} + a_6 \lambda^2 \quad (5.2)$$

where $f = (T - 24.5)(T + 570.82)$, and $a_1, a_2, a_3, a_4, a_5, a_6, b_1, b_2, b_3, b_4$ are the coefficients of a magnesium doped periodically poled lithium niobate (MgO:PPLN) crystal from Deltronic crystals [24], a quasi-phase-matched material. Quasi-phase-matching allows the use of nonlinear crystals that would otherwise not be phase-matched. This is achieved by the periodic polling of the crystal.

The polling in the crystal is to allow continuous buildup of the newly generated photons by using alternating electric dipole orientations for constructive interference as light propagates within the crystal [25]. Adjusting the polling periodicity with temperature and wavelength ensures maximum conversion efficiency. The results of combining Equations 5.1 and 5.2 to estimate the phase matching temperature is shown in Figure 16. The results allowed for a theoretical approximation of the working temperature and its rate of change with respect to varying wavelength and polling periodicity. While this is a good approximation, an exhaustive temperature analysis allowed me to find the exact temperature for maximum second harmonic generation. The experimental optimal temperature of 191.6 C from Figure 17b, matched closely with the theoretical approximation of 195.1 C for a 20 mW laser diode source at 1560 nm. There is a 5.9 C/nm rate of change of the temperature with varying wavelength. This mathematical model is also consistent with a previous CW source used, which was a 5 mW tunable source at 1560.38 nm. Its theoretical phase-matching temperature was 197.4, and its

experimental phase-matching temperature at this wavelength was 194.4 C. It is also shown in Figure 17 that the deviation from the phase matching temperature caused a decrease in second harmonic generation and spectral sidebands.

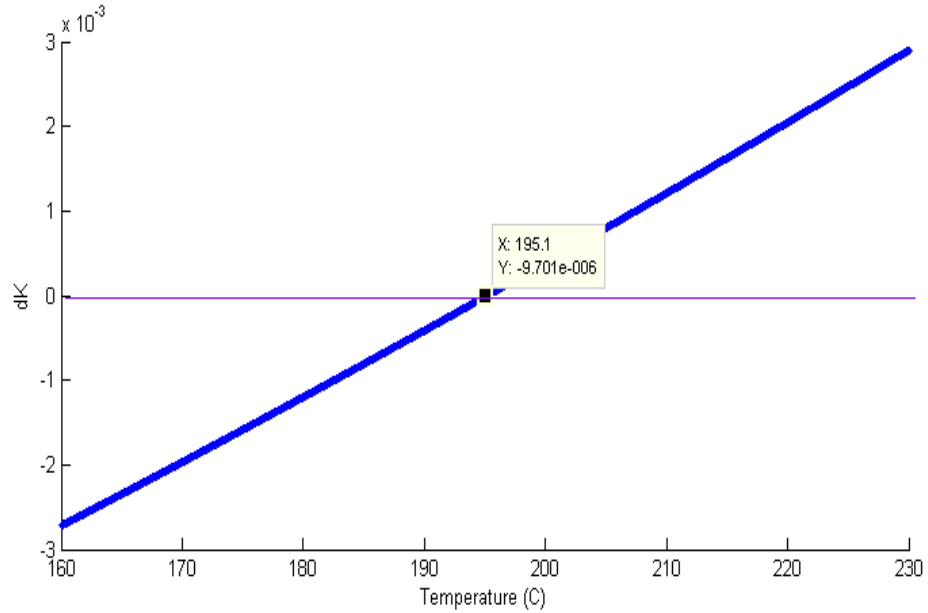


Figure 16. Mathematical model result for finding the phase-matching temperature using the Sellmeier equation and phase-matching condition relation.

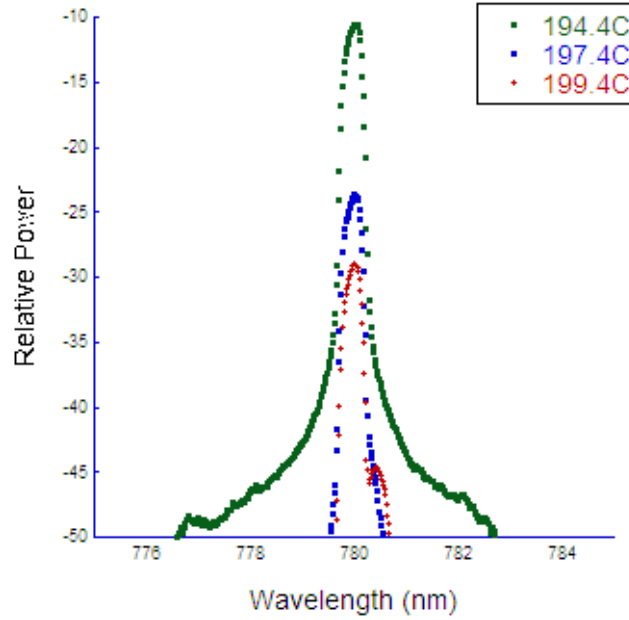


Figure 17a. Peak Power spectral response of 780 nm output to temperature variation using the 5 mW tunable 1560.38 nm CW source

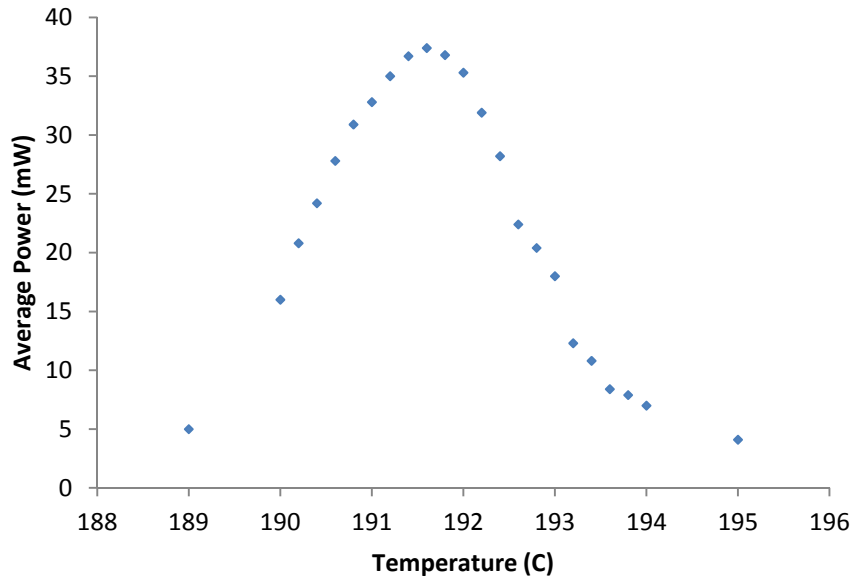


Figure 17b. Measured 780 nm power with respect to varying PPLN oven temperature. The size of the marker is a measure of the uncertainties.

Crystal alignment

Operation in the near infrared (NIR) regime makes it difficult to align the beam into the conversion crystal. The crystal had six domains, each with a different polling period, Λ , the widest of which has $\Lambda = 19\mu\text{m}$. I used a 632.8 nm HeNe laser in the visible red to locate the midpoint of the widest domain. I translated the crystal both vertically and horizontally and recorded the occurrences of small anomalies in the laser beam profile. These anomalies occurred due to reflection when the beam passed over domain boundaries. This method resulted in an efficient and accurate way of both aligning the crystal and calculating the midpoints of its domains. I also did an iterative experiment of varying the focal length of the focusing lens and adjusted its distance from the input facet of the crystal to control the beam spot size.

This process varies the beam spot size, for optimal focusing and efficient second harmonic generation.

Initial power measurements of the pulsed 1560 nm source

There were two ways in which the power output of the 1560 nm light from the pulse-carving/EDFA system was measured: using a thermal meter at the 99% (see Figure 15) output and using an oscilloscope at the attenuated 1% output. The average power obtained from the thermal meter was multiplied by the duty cycle to calculate the peak power. The measured attenuated 1% 1560 nm signal's peak power is calculated by multiplying by the attenuating factor. Both measurements were made and shown in Figure 18 for each repetition rate starting at 19 MHz, doubled each cycle, to 1.25 GHz, while maintaining a pulse duration of 50 ps at the FWHM.

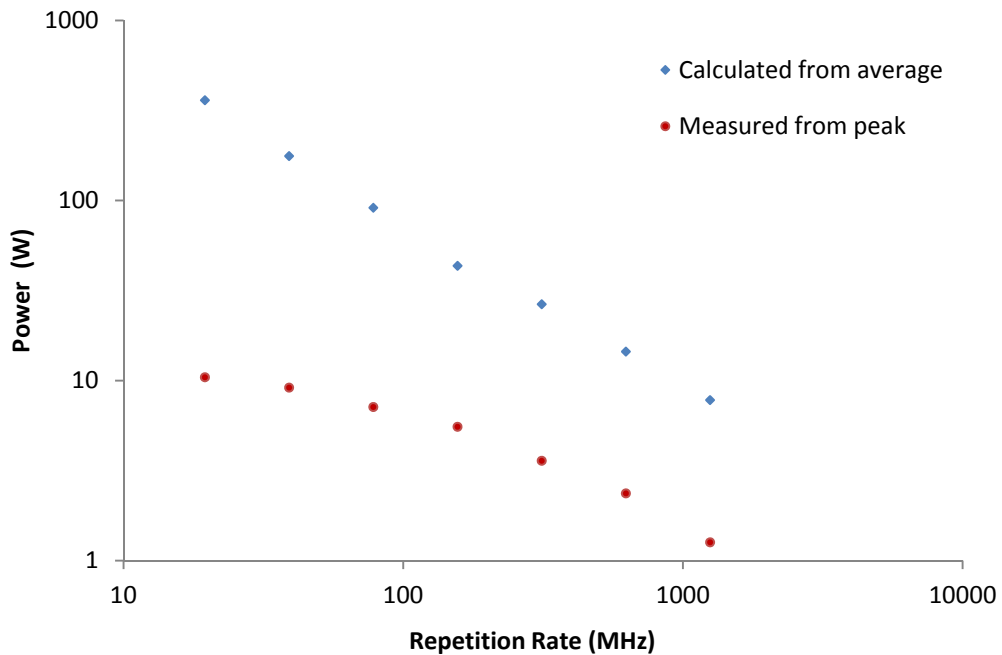


Figure 18. A comparison of peak power measurement from an attenuated input to the oscilloscope and calculated from average power using a tunable 1560 nm CW source. The size of the marker is a measure of the uncertainties.

There is a noticeable power difference between the two curves seen in Figure 18. In a perfect system, these curves would coincide with each other. The discrepancy can be attributed to uncalibrated attenuators and low input power to the EDFA. There are also differences between the curve measured from the peak and the curve calculated from the average in the low repetition rate regime. This effect can be ascribed to amplified spontaneous emission (ASE). ASE is a process by which spontaneous emission in an optical amplifier is then amplified by stimulated emission and has a major impact on the available gain. As the interval between input pulses becomes longer, amplification of the ASE draws more power from the EDFA. Because ASE can occur anywhere in the gain bandwidth, measurement of suspected ASE was done by using the attenuated 1% signal as input to an optical spectrum analyzer. The presence of the ASE was detected using this method, as illustrated in Figure 19. The purple curve is the spectral response of the EDFA without any input signal attached. It is also clear from the result that the contribution of the ASE in the signals becomes larger as the repetition rate decreases.

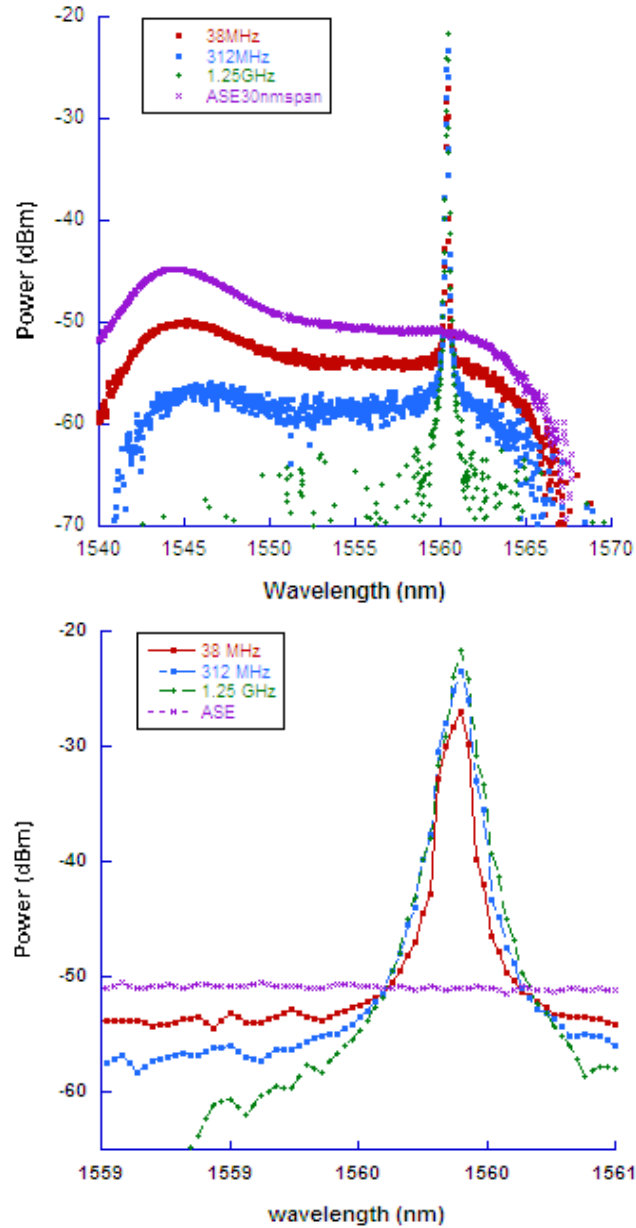


Figure 19. Top: Spectral response from varying repetition rate to observe presence of amplified spontaneous emission(ASE) from the EDFA Bottom: Spectral response around the maximum

The 5 mW tunable 1560 nm CW source was replaced by a 20 mW 1560 nm narrow line laser diode to reduce the effect of the ASE. Theoretically, the more input power available for driving the EDFA, the less the amplification that will be drawn by the ASE. There is no ASE observed with decreasing repetition rate in the spectral

analysis with the new 1560 nm source as seen in Figure 20. The same peak power measurements were performed to determine the power response of the laser diode. The results in Figure 21 show that, as expected, the new laser diode resulted in higher 1560 nm peak power measurements than the tunable laser. However, there is a divergence between the two curves at the lower repetition rate regime, despite the new laser diode source. The difference in the divergence was approximated by measuring the slope of the secant line through the 19MHz and 77MHz data points. The slope of the line is -0.0563 watt per hertz in Figure 18 and -0.2375 watt per hertz in Figure 21. This suggests a reduction in the undesired divergence.

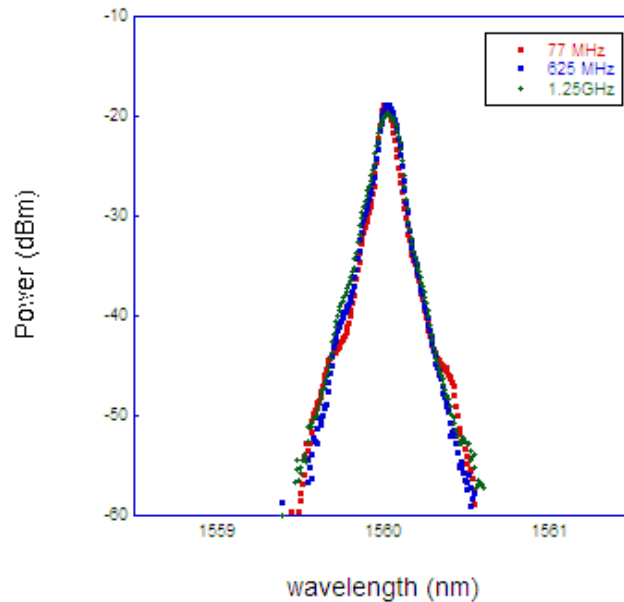


Figure 20. Spectral response of varying repetition rate without the presence of ASE

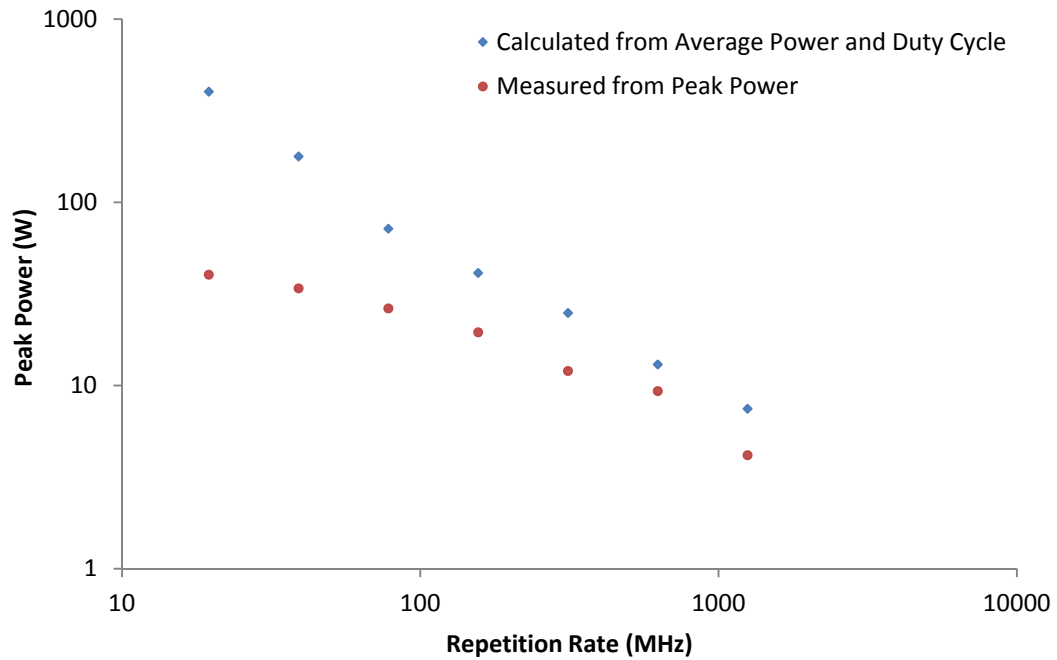


Figure 21. A comparison of peak power measurement from an attenuated input to the oscilloscope and calculated from average power using a laser diode source. The size of the marker is a measure of the uncertainties.

Maximizing the 780 nm peak power output

Intuitively, more 1560 nm input power to the conversion crystal is desired in order to increase the 780 nm throughput. Since the input polarization into the EDFA affects the output power due to the polarization-dependent gain [23], maximization of either the 1560 nm or 780 nm power is done by controlling the polarization paddles at the input and output of the EDFA. The data points from the blue curve in Figure 21 were obtained by measuring the 1560 nm input to the conversion crystal using a thermal detector while maximizing the 1560 nm input. Conversely, the data points from the blue curve in Figure 22 were obtained by measuring the 780 nm output of the conversion crystal using a thermal detector while maximizing the 780 nm output.

Unexpectedly, there was no clear relation between the maximized 1560 nm input and 780 nm output. The overlap between the two curves of 780 nm power measurements starts to deviate at 77 MHz as seen in Figure 23. However, the difference between the peak power calculated using a thermal detector and the oscilloscope almost disappeared. At this time, the precise cause of this phenomenon is unclear, but it is polarization related suggested by the polarization dependence in maximizing the peak power. Nonetheless, the versatile 780 nm pump source was able to produce 18.5 W peak power at 19 MHz and 217 mW peak power at 1.25 GHz. The available 780 nm power is ample for the next stage of this overall project.

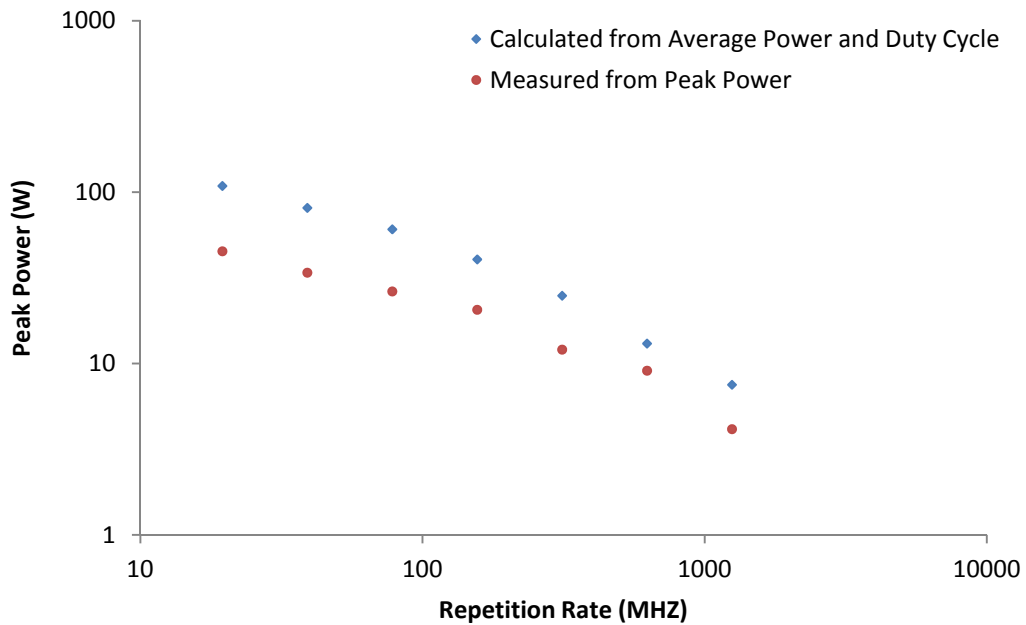


Figure 22. 1560 nm peak power measurement at maximized 780 nm output with polarization control. The size of the marker is a measure of the uncertainties

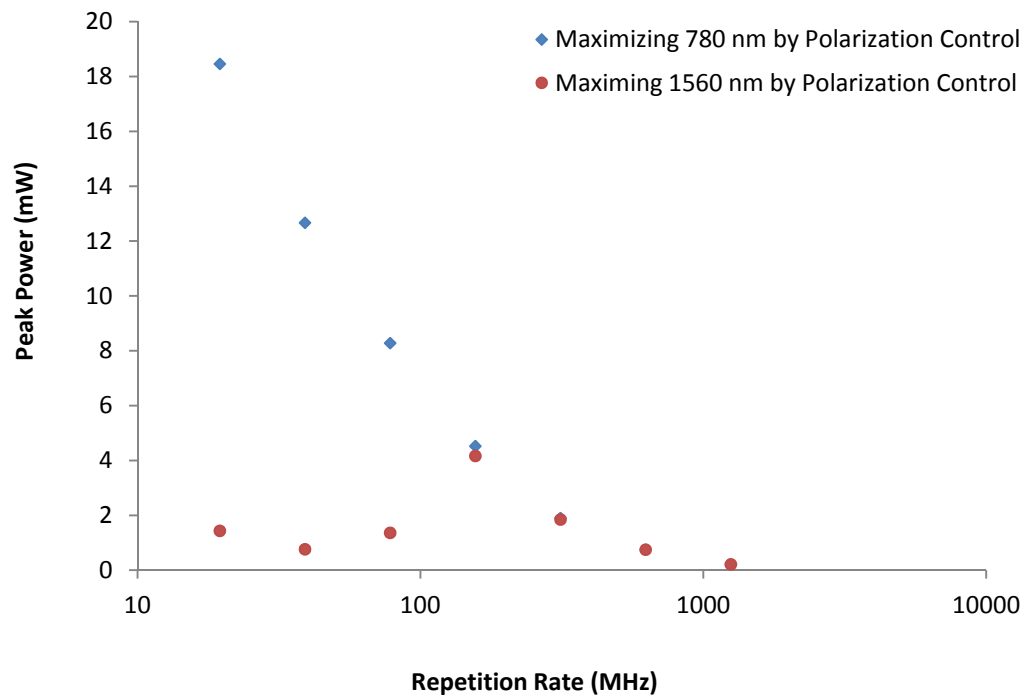


Figure 23. 780 nm peak power measurement compared to maximized 1560 nm radiation. The size of the marker is a measure of the uncertainties.

Chapter 5: High-speed generation of correlated photon pairs

What can we do with it?

The theory of four-wave mixing in a photonic crystal fiber (PCF), also referred to as microstructure fiber, allows for the emission at 656.28 nm, which is also referred to as the H_{α} line. A microstructure fiber is a special type of optical fiber that has a specific arrangement of closely-spaced air holes throughout the length of the fiber. The most common configuration is a triangular array with one air hole missing. That region has a higher refractive index that acts just like the core in a conventional optical fiber (refer to figure 24).

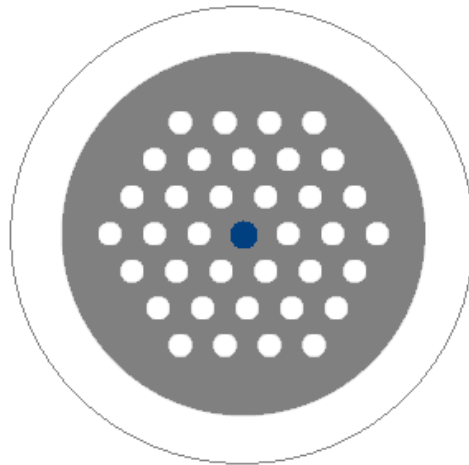


Figure 24. Microstructure fiber structure with 2.3 μm core diameter.

Operation at the H_{α} wavelength reduces the background noise in a free space optical QKD channel, consequently increasing the efficiency. Therefore, a high-speed entangled photon pair generator producing a stream of photons at 656.28 nm

wavelength is a desirable source for a quantum telecommunication network. Since we have a 780 nm pump source, and we want to produce signal photon at the 656 nm wavelength, we can calculate the wavelength of the idler photon using,

$$\frac{2}{\lambda_p} = \frac{1}{\lambda_s} + \frac{1}{\lambda_i} \quad (6.1)$$

where λ_p is the pump photon wavelength and λ_s , λ_i are the signal and idler wavelength respectively. This is a result of the interaction between the photons in the incident optical field with the particles of matter in the nonlinear medium. The difference between the vibrational and rotational energy of the molecules inside the crystal from the scattering of light will cause a nonlinear response of photons with shifted energy. Two photons of the pump field in the $\chi^{(3)}$ nonlinear medium can create a pair of signal and idler photons. This is done by injecting an optical pump field into the microstructure fiber near the zero dispersion region [26]. By convention, the frequency of the signal photon is greater than the pump frequency, while the frequency of the idler photons is less than the pump frequency. Therefore, for a 780 nm input pump field wavelength, the signal and idler wavelengths will be 656 nm and 961 nm respectively of equation 6.1.

Experimental layout of correlated pair generation

The versatile 780 nm pump source from the Chapter 4 was used for the correlated pair generator shown in Figure 24. A narrow-line filter is to eliminate unwanted 1560 nm radiation. The objective lenses before and after the PCF are used to focus and collect light respectively. Another dichroic mirror is used to separate the 656 nm and 961 nm components. Notch filters were used to eliminate the 780 nm

radiation exiting of the microstructure fiber. The long wave pass (LWPF) and the short wave pass (SWPF) filters is to allow transmission of wavelengths above and below 800 nm. Lastly, there is a 10 nm band pass filter (BPF) in each of the beam paths in front of the detector to reflect all spurious light. It is important to remove as much 780 nm light as possible and have a clean signal of 656 nm in the transmitted optical path and 961 nm in the reflected optical path, respectively. Three 60 dB notch filters were placed in series, which effectively extinguishes 180 dB of 780 nm radiation to achieve desired clean signal of 656 nm and 961 nm. The coupling efficiency of the 780 nm pump source into the microstructure fiber is 50%.

The outputs from the two detectors were used as inputs to a time-correlated single photon counting card. The counting card performs time-to-amplitude conversion (TAC) and analog-to-digital conversion (ADC) of the time between detection events. If there is residual unfiltered 780 nm radiation, the detection events that correspond to that wavelength contribute to the background noise. The output from the thin detector acts as the start signal, and the output from the thick detector acts as the stop signal.

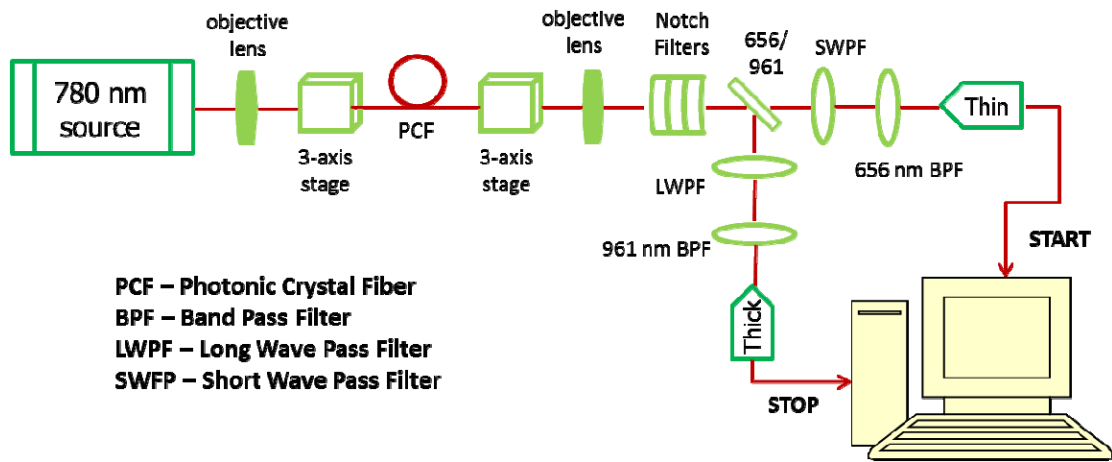


Figure 25. Schematic of high speed correlated pair generator [27]

Coincidence measurements

There is about 90% transmission loss for both the signal and idler beam due to the optical components such as the filters along the propagation path to the detectors. As a result, it is easy to lose the correlated pairs generated before they reach the detector. Shown in Table 1 is the percentage of transmission of each optical element. Moreover, even if these photons reach the detector surface, the detection efficiencies at the wavelengths of interest are less than 40%. Currently, we observe accidental coincidences for 656 nm and 961 nm wavelengths at 38 MHz. The validation is due to the 180 dB attenuation of the 780 nm from the three notch filters and the 10 nm bandwidth band pass filters for both the 656 nm and 961 nm wavelengths. Moreover, the photon counts from the detector do not linearly increase with increasing microstructure fiber output. This suggests nonlinear phenomenon in the fiber, which is attributed to four-wave mixing.

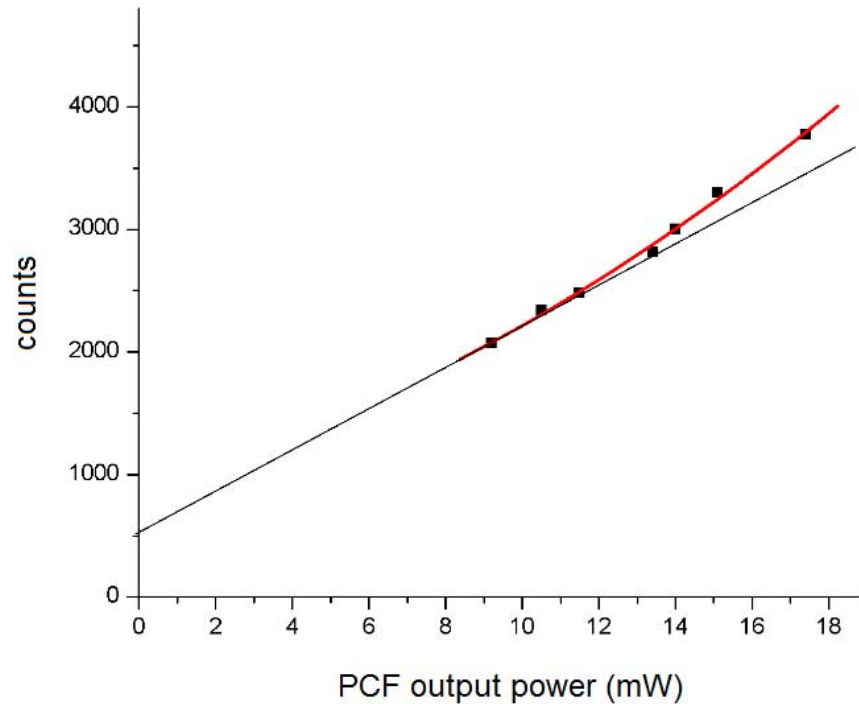


Figure 26. Nonlinear relationship between PCF output power and count rate in the 656.28 nm beam path. Statistical uncertainties in the counts are smaller than the marker. It grows by a factor of \sqrt{N} , where N is the count.

Proof of correlated pair generation in the coincidence experiments has not yet been achieved. This is most likely due to the misaligned and unfocused beam into the detector surface or low actual nonlinear specifications of the microstructure fiber. There is only about 6.7% probability that a 961 nm photon will be detected and about 11% probability that a 656 nm photon will be detected as described in Table 1. It is even more difficult to detect the correlated photon pair at the same time.

Elements	% Transmission of 961 nm	% Transmission of 656 nm
Detection efficiency	20	35
BPF	55	55
MMF coupling	86	82
LWPF	85	n/a
SWPF	n/a	90
Dichroic mirror	97	98
Notch filter	95	95
Total transmitted	7%±3%	11%±3%

Table 1. List of transmission percentage of optical elements along the wave propagation axis of the correlated photon pair generator

Chapter 6: Conclusion

Future development

Efficient high-speed photon entanglement generators that are compatible with our QKD system are not easily available in the market. Even the desired Ti sapphire laser source supports up to 100MHz generation only. The components of the entanglement generator are shown in Figure 26. Once the generator is constructed, the 656 nm signal can be sent over the free space link

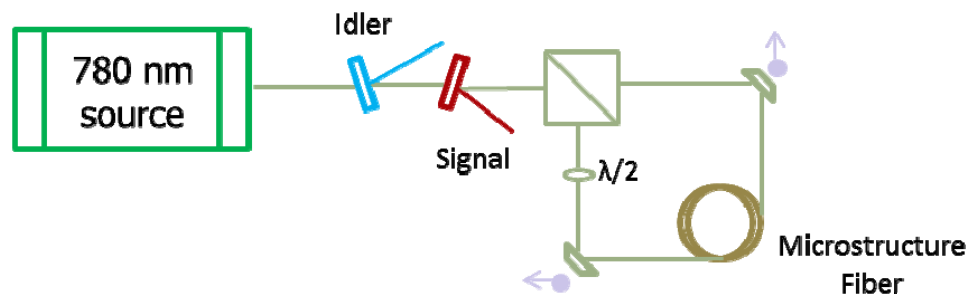


Figure 27. Schematic of the entanglement generator using a microstructure fiber [28]

Lessons learned

Laser safety and quantum telecommunications merely scratch the surface of the myriad of applications of sum-frequency generation. Understanding the basic underlying concept of SHG allowed us to discover hidden hazards associated with commonly the used green laser pointer and alert the public. The knowledge that green radiation is obtained by frequency doubling 1064 nm infrared radiation led to

the hypothesis regarding a possible infrared leakage. No infrared-blocking was found near the output of the laser pointer to contain possible infrared leakage. An experiment using common household items to effectively display the presence of unwanted infrared radiation.

A more technical understanding of the same concept of SHG was applied to work in the broadband telecommunications regime. It resulted in a versatile source of 780 nm pulses, capable of operating at a repetition rate of 1.25 GHz, suitable for use in a correlated pair generator. The proposed generator is an improvement over the current operational QKD system, that will ultimately protect the confidentiality of private cyber conversations and personal information.

Appendix A

Graphical user interface to visually display and monitor secret key production and error rates of the portable QKD system via TCP/IP

Code written in CVI LabWindows

//file: rates.cws

```
#include <formatio.h>
#include <utility.h>
#include <userint.h>
#include <cvirte.h>
#include <stdio.h>
#include <string.h>
#include <tcpsupp.h>
#include <ansi_c.h>
#include <analysis.h>
#include "Rate.h"
#define tcpChk(f) if ((g_TCPError=(f)) < 0) {ReportTCPError(); goto Done;} else
#define AVG_N 3

int CVICALLBACK ClientTCPCB (unsigned handle, int event, int error,
                             void *callbackData);
int CVICALLBACK Destination (unsigned handle, int event, int error,
                             void *callbackData);
static void ReportTCPError (void);
unsigned long int my_ntohl(unsigned long int);
unsigned long int my_average(unsigned long int);
static int          mainPNL;
static unsigned int g_hconversation;
static int          g_connected = 0;
static int          g_TCPError = 0;
static int          j;
static unsigned long int dataArray[AVG_N] = {0};
static int          index = 0;

int main (int argc, char *argv[])
{
    // load the Panel
    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;
    if ((mainPNL = LoadPanel (0, "Rate.uir", PANEL)) <= 0)
        goto Done;
}
```



```

    DisableBreakOnLibraryErrors();
        DisplayPanel (mainPNL);
    RunUserInterface ();
Done:
    /* Disconnect from the TCP server */
    if (g_connected)
        DisconnectFromTCPSTerver (g_hconversation);

    /* Free resources and return */
    DiscardPanel (mainPNL);
    CloseCVIRTE ();
    return 0;
}
//Connects to the server after the start button click and turns on the LED
int CVICALLBACK start(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2)
    {
        int portNum = 5071;
        //char tempBuf[256] = "129.6.141.28";
        char tempBuf[256] = "129.6.41.26";

        if (event == EVENT_COMMIT){
            SetWaitCursor (1);
            if (ConnectToTCPSTerver (&g_hconversation, portNum, tempBuf,
Destination,
                NULL, 5000) < 0) {
                MessagePopup("TCP Client", "Connection to server failed !");
            }
            else
            {
                SetWaitCursor (0);
                g_connected = 1;
                SetCtrlVal (PANEL, PANEL_LED, 1);
                ClientTCPCB(g_hconversation, TCP_DATAREADY, 0,0);
            }
        }
        return 0;
    }

int CVICALLBACK ClientTCPCB (unsigned handle, int event, int error,
void *callbackData)
    {
        unsigned long int received;
        unsigned long int shake;
        unsigned long int sendData = 4;
        unsigned long int siftedData;

```

```

double Ksifted;
unsigned long int errorData;
unsigned long int secretData;
double Ksecret;
unsigned long int retainRatio;
unsigned long int newSiftedData;
double newErrorData;
unsigned long int newSecretData;
unsigned long int newRetainRatio;
unsigned long int handshake;
unsigned long int data_formatted[4];
unsigned long int ave_sifted;
unsigned long int ave_secret;
double ave_error;
int stop;
double t;
j=1;
switch(event) {

case TCP_DATAREADY:

ClientTCPRead(g_hconversation, &received, 4, 2000);
shake = my_ntohl(received);
//SetCtrlVal (PANEL, PANEL_hands, shake);
// MessagePopup("JEM", "RECEIVED");

// Checks for the proper protocol from server to continue otherwise illegal
connection
//if (shake != 0) {
// MessagePopup("TCP CLIENT", "Server Acknowledgement Error");
// goto Done;
//}
//else {
//for(j=1; j<4; j++) {
while (1) {

t = Timer ();
while (Timer () - t < 12.0) {
ProcessSystemEvents ();
GetCtrlVal (PANEL, PANEL_STOPBUTTON,
&stop);

if (stop) {
break; // Exit the pause loop
}
}
if (stop){
QuitUserInterface (0);

```

```

        break; // Exit the test loop
    }

    ClientTCPWrite (g_hconversation, &sendData, 4, 2000);
    //MessagePopup("TCP Client", "data sent");

    // Reads the incoming four sets of 4-bytes of
information
    ClientTCPRead (g_hconversation, &data_formatted,
16, 2000);

    siftedData = data_formatted[0];
    newSiftedData = my_ntohl(siftedData);
    ave_sifted = my_average(newSiftedData);
    Ksifted = (1.0*newSiftedData)/(1.0*1000);

    errorData = data_formatted[1];
    newErrorData = (1.0*(my_ntohl(errorData)))/(1.0*10);
    //ave_error = (1.0*(my_average(newErrorData)))/(10*1.0);

    secretData = data_formatted[2];
    newSecretData = my_ntohl(secretData);
    ave_secret = my_average(newSecretData);
    Ksecret = (1.0*newSecretData)/(1.0*1000);

    retainRatio = data_formatted[3];
    newRetainRatio = my_ntohl(retainRatio);

    PlotStripChart (PANEL,
PANEL_SIFTED, &Ksifted, 1, 0, 0,
    VAL_DOUBLE);
    //var = (int)siftedData;
    SetCtrlVal (PANEL, PANEL_SIFTEDNUMBER,
Ksifted);

    PlotStripChart (PANEL, PANEL_ERROR, &newErrorData, 1,
0, 0,
    VAL_DOUBLE);
    SetCtrlVal (PANEL, PANEL_ERRORNUMBER,
newErrorData);

    PlotStripChart (PANEL, PANEL_SECRET, &Ksecret, 1, 0, 0,
    VAL_DOUBLE);
    SetCtrlVal (PANEL, PANEL_SECRETNUMBER,
Ksecret);

```

```

        SetCtrlVal (PANEL, PANEL_RATIO,
newRetainRatio);
    }

    // }
    // }
    break;
    case TCP_DISCONNECT:
        MessagePopup ("TCP Client", "Connection Terminated");
        g_connected = 0;
        MainPanelCB (0, EVENT_CLOSE, 0, 0, 0);
        break;
    }

Done:
    if (g_connected) {
        DisconnectFromTCPSTerver (g_hconversation);
        MessagePopup ("TCP Client", "Connection Terminated");
        SetCtrlVal (PANEL, PANEL_LED, 0);
    }
    return 0;
}

int CVICALLBACK Destination (unsigned handle, int event, int error,
void *callbackData)
{
    return 0;
}

//Function to rearrange the bit order and retrieve original
unsigned long int my_ntohl(unsigned long int in){
    unsigned long int nib, out;
    int i;
    out=0;
    for (i=0; i<32; i=i+8){
        nib = (in>>i)&0x000000FF;
        out = ((out<<8)&0xFFFFFFFF0)|nib;
    }
    return out;
}

unsigned long int my_average(unsigned long int input){
    int output = 0;
    int z=0;

```

```

    int ave;
    dataArray[index] = input;
    index++;
    if (index == AVG_N){
        index = 0;
    }
    for(z=0; z<AVG_N; z++){
        output = output + dataArray[z];
    }

    ave = output/AVG_N;
    return ave;
}

int CVICALLBACK MainPanelCB (int panel, int event, void *callbackData,
                             int eventData1, int eventData2)
{
    //j=0;
    if (event == EVENT_CLOSE)
        QuitUserInterface (0);
    return 0;
}

int CVICALLBACK Reset(int panel, int control, int event, void *callbackData, int
eventData1, int eventData2)
{
    //j=0;
    if (event == EVENT_COMMIT)
    {
        ClearStripChart (PANEL, PANEL_SIFTED);
        ClearStripChart (PANEL, PANEL_ERROR);
        ClearStripChart (PANEL, PANEL_SECRET);
        SetCtrlVal (PANEL, PANEL_RATIO, 0);
        SetCtrlVal (PANEL, PANEL_SECRETNUMBER, 0.0);
        SetCtrlVal (PANEL, PANEL_ERRORNUMBER, 0.0);
        SetCtrlVal (PANEL, PANEL_SIFTEDNUMBER, 0.0);
    }
    return 0;
}
/*
int CVICALLBACK QuitCallback(int panel, int control, int event, void
*callbackData, int eventData1, int eventData2)
{
    //j=0;

```

```
    if (event == EVENT_COMMIT)
        QuitUserInterface (0);
    return 0;
} */

static void ReportTCPErrors (void)
{
    char messageBuffer[1024];

    if (g_TCPErrors < 0)
    {
        sprintf(messageBuffer,
            "TCP library error message: %s\nSystem error message: %s",
            GetTCPErrorsString (g_TCPErrors), GetTCPSystemErrorsString());
        MessagePopup ("Error", messageBuffer);
        g_TCPErrors = 0;
    }
}
```

Appendix B

Graphical user interface to compare repeatability and predictability of TRNGs vs.

PRNGs

Code written in CVI LabWindows

//file: rng.cws

```
#include <tcpsupp.h>
#include "toolbox.h"
#include <analysis.h>
#include <formatio.h>
#include <cvirte.h> /* Needed if linking in external compiler; harmless otherwise */
#include <utility.h>
#include <userint.h>
#include <stdio.h>
#include <stdlib.h>
#include <ansi_c.h>
#include <math.h>
#include "rng.h"

#define COUNT 100
#define COUNT2 200
#define max 10000 /* number of steps */
#define Sqrt_2 1.4142135623730950488
#define tcpChk(f) if ((g_TCPError=(f)) < 0) {ReportTCPError(); goto Done;} else

int CVICALLBACK ClientTCPCB (unsigned handle, int event, int error,
                             void *callbackData);
int CVICALLBACK Destination (unsigned handle, int event, int error,
                              void *callbackData);

static void ReportTCPError (void);
unsigned long int my_ntohl(unsigned long int);

FILE *output; /* internal file name */

static unsigned int g_hconversation;
static int g_connected = 0;
static int g_TCPError = 0;
static int panelHandle, panelHandle2;
static int stop, timer;
static int maxxIndex, minxIndex, maxyIndex, minyIndex;
```

```

static int maxxIndex2, minxIndex2, maxyIndex2, minyIndex2;
static double wave[100], wave1[100], array[100], array1[100], xonea[10001],
yonea[10001],xoneb[10001], yoneb[10001];
static double newArray[3400], fileArray[3400];
static double maxxVal, minxVal, maxyVal, minyVal;
static double maxxVal2, minxVal2, maxyVal2, minyVal2;
static char binary[3400];
static char proj_dir[MAX_PATHNAME_LEN];
static char file_name[MAX_PATHNAME_LEN];
static char file_name1[MAX_PATHNAME_LEN];
static double seed,seed2;
int hist_array[2048], y_array[2048];
double temp_history1[512], x_array[4096];
unsigned long int temp_history[2048];

```

```

int main (int argc, char *argv[])
{
    GetProjectDir (proj_dir);

    if (InitCVIRTE (0, argv, 0) == 0)
        return -1;
    if ((panelHandle = LoadPanel (0, "rng.uir", PANEL)) <= 0)
        goto Done;
    DisableBreakOnLibraryErrors();
    DisplayPanel (panelHandle);
    RunUserInterface();

Done:
    /* Disconnect from the TCP server */
    if (g_connected)
        DisconnectFromTCPServer (g_hconversation);

    /* Free resources and return */
    DiscardPanel (panelHandle);
    CloseCVIRTE ();
    return 0;
}

```

```

int CVICALLBACK StartCallback (int panel, int control, int event, void
*callbackData,
    int eventData1, int eventData2)
{
    int portNum = 5071;

```



```

char tempBuf[256] = "129.6.141.20";
double x,y,k,l,m,num,err,w;
int i, points;
if (event == EVENT_COMMIT){

    SetWaitCursor (1);
    if (ConnectToTCPServer (&g_hconversation, portNum, tempBuf,
Destination,
        NULL, 5000) < 0) {
        MessagePopup("TCP Client", "Connection to server failed !");
    }
    else
    {
        SetWaitCursor (0);
        g_connected = 1;
        ClientTCPCB(g_hconversation, TCP_DATAREADY, 0,0);
    }
    /*----- QRNG -----*/

    if (FileSelectPopup ("", "*.dat", "*.dat", "Name of File to Read",
        VAL_OK_BUTTON, 0, 0, 1, 0, file_name) > 0){
        GetCtrlVal (panelHandle, PANEL_InputType, &fileType);

        FileToArray (file_name, array, VAL_DOUBLE, COUNT, 1,
            VAL_GROUPS_TOGETHER,
VAL_GROUPS_AS_COLUMNS, fileType);
        Copy1D (array, COUNT, xonea);
            Copy1D (array + COUNT, COUNT, yonea);

            DeleteGraphPlot (panelHandle, PANEL_GRAPH_3, -
1, 1);
            PlotXY (panelHandle, PANEL_GRAPH_3, xonea, yonea, COUNT,
VAL_DOUBLE, VAL_DOUBLE,
            VAL_THIN_LINE, VAL_NO_POINT, VAL_SOLID,
1,VAL_BLUE);
        } */
    }

    return 0;
}

int CVICALLBACK Destination (unsigned handle, int event, int error,
    void *callbackData){
    return 0;
}

```

```

int CVICALLBACK ClientTCPCB (unsigned handle, int event, int error,
                             void *callbackData){
    int i,j,d, seeding;
    double t,x, y, a, b;
    unsigned long int received;
    unsigned long int shake, shake1;
    unsigned long int sendData = 4;
    double norm = 2147483648;
    unsigned long int randplotx[400];
    unsigned long int randplotx_new[400];
    double walk_x1[400];
    double walk_x2[400];
    unsigned long int randploty[400];
    unsigned long int randploty_new[400];
    double walk_y1[400];

    double walk_y2[400];
    unsigned long int histo[2048];
    unsigned long int raw[2048];
    double raw_new[2048];
    unsigned long int histo1[1024];
    unsigned long int whiten[1024];
    double whiten_new[1024];

    switch(event) {

case TCP_DATAREADY:

    ClientTCPRead(g_hconversation, &received, 4, 5000);
    shake = my_ntohl(received);
    //SetCtrlVal(panelHandle, PANEL_COLORNUM_3, shake);
    DisplayImageFile (panelHandle, PANEL_PICTURE,
"randbitmap_true.bmp");
        DisplayImageFile (panelHandle, PANEL_PICTURE_2,
"randbitmap_computer.bmp");
        while (1){
            t = Timer ();
            while (Timer () - t < 2.0) {
                ProcessSystemEvents ();
                GetCtrlVal (PANEL, PANEL_STOPBUTTON,
&stop);
                if (stop) {

                    break;

```

```

    }

    }
    if (stop){
        QuitUserInterface (0);
        break;
    }
ClientTCPWrite (g_hconversation, &sendData, 4, 5000);

/* ----- QRNG Walk -----
-----*/

//Read 800 bytes for 100 Data Points for random walk plot 1
ClientTCPRead(g_hconversation, &randplotx, 400, 5000);
ClientTCPRead(g_hconversation, &randploty, 400, 5000);
for (j=0; j<100; j++){
    randplotx_new[j] = my_ntohl(randplotx[j]);

    walk_x1[j] = (randplotx_new[j]);
    randploty_new[j] = my_ntohl(randploty[j]);
    walk_y1[j] = (randploty_new[j]);
}
ArrayToFile ("jem0.txt", randplotx_new, VAL_INTEGER, 100, 1,
VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
10, VAL_ASCII, VAL_TRUNCATE);
ArrayToFile ("jem1.txt", walk_x1, VAL_DOUBLE, 100, 1,
VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
10, VAL_ASCII, VAL_TRUNCATE);
ArrayToFile ("jem2.txt", randploty_new, VAL_INTEGER, 100, 1,
VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
10, VAL_ASCII, VAL_TRUNCATE);
ArrayToFile ("jem3.txt", walk_y1, VAL_DOUBLE, 100, 1,
VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
10, VAL_ASCII, VAL_TRUNCATE);
//Read 800 bytes for 100 Data Points for random walk plot 2
ClientTCPRead(g_hconversation, &randplotx, 400, 5000);
ClientTCPRead(g_hconversation, &randploty, 400, 5000);

for (j=0; j<100; j++){
    randplotx_new[j] = my_ntohl(randplotx[j]);
    walk_x2[j] = (randplotx_new[j]);

```

```

        randploty_new[j] = my_ntohl(randploty[j]);
        walk_y2[j] = (randploty_new[j]);
    }

    x = COUNT/2;
    y = x;
    for (i=1; i<COUNT; i++){
        d = i-1;
        /*x = x + (walk_x1[i] - 0.5) * 3;
        y = y + (walk_y1[i] - 0.5) * 3;
        wave[i] = x;
        array[i] = y;    */
        x = x + (walk_x1[d] - 1073741824) * SQRT_2;
    y = y + (walk_y1[d] - 1073741824) * SQRT_2;
    wave1[i] = x/10000000;
    array1[i] = y/10000000;
    }
    DeleteGraphPlot (panelHandle, PANEL_GRAPH_3, -1, 1);

    for (i=1; i<COUNT; i++){
        d=i-1;
        MaxMin1D (wave1, COUNT, &maxxVal,
&maxxIndex, &minxVal, &minxIndex);
        MaxMin1D (array1, COUNT, &maxyVal, &maxyIndex, &minyVal,
&minyIndex);

        SetAxisScalingMode (panelHandle,
PANEL_GRAPH_3, VAL_BOTTOM_XAXIS,
        VAL_MANUAL, minxVal, maxxVal);
        SetAxisScalingMode (panelHandle, PANEL_GRAPH_3,
VAL_LEFT_YAXIS,
        VAL_MANUAL, minyVal, maxyVal);
        PlotLine(panelHandle, PANEL_GRAPH_3, wave1[d],
array1[d], wave1[i], array1[i], VAL_MAGENTA);
        SetCtrlAttribute (panelHandle, PANEL_TIMER,
ATTR_ENABLED, 1);
        //Delay(.02);
    }
    x = COUNT/2;
    y = x;
    for (i=1; i<COUNT; i++){
        d = i-1;
        /*x = x + (walk_x1[i] - 0.5) * 3;
        y = y + (walk_y1[i] - 0.5) * 3;
        wave[i] = x;
        array[i] = y;    */
        x = x + (walk_x2[d] - 1073741824) * SQRT_2;

```

```

y = y + (walk_y2[d] - 1073741824) * SQRT_2;
wave1[i] = x/10000000;
array1[i] = y/10000000;
}
DeleteGraphPlot (panelHandle, PANEL_GRAPH_4, -1, 1);
for (i=1; i<COUNT; i++){
    d=i-1;
    MaxMin1D (wave1, COUNT, &maxxVal,
&maxxIndex, &minxVal, &minxIndex);
    MaxMin1D (array1, COUNT, &maxyVal, &maxyIndex, &minyVal,
&minyIndex);
    SetAxisScalingMode (panelHandle,
PANEL_GRAPH_4, VAL_BOTTOM_AXIS,
VAL_MANUAL, minxVal, maxxVal);
    SetAxisScalingMode (panelHandle, PANEL_GRAPH_4,
VAL_LEFT_AXIS,
VAL_MANUAL, minyVal, maxyVal);
    PlotLine(panelHandle, PANEL_GRAPH_4, wave1[d],
array1[d], wave1[i], array1[i], VAL_MAGENTA);
    //Delay(.02);
}
/*----- rand() -----*/

GetCtrlVal (panelHandle, PANEL_BINARYSWITCH, &seeding);
GetCtrlVal (panelHandle, PANEL_COLORNUM, &seed);

if (seeding == 1) {
    //srand(seed);
    SetRandomSeed(seed);
}
x = COUNT/2;
y = x;
DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, 1);
for (i=1; i<COUNT; i++){
    ProcessSystemEvents ();
    GetCtrlVal (panelHandle, PANEL_STOPBUTTON,
&stop);
    if (stop == 1){
        ResetTimer (panelHandle, PANEL_TIMER);
        SetCtrlAttribute (panelHandle,
PANEL_TIMER, ATTR_ENABLED, 0);
        break;
    }
    a = ((Random(0,1)) - 0.5) * 100;
    b = ((Random(0,1)) - 0.5) * 100;
    x = x+a;

```

```

        y = y+b;
        wave[i] = x;
        array[i] = y;
    }
        for (i=1;i<COUNT;i++){
            d = i-1;
            MaxMin1D (wave, COUNT, &maxxVal, &maxxIndex, &minxVal,
&minxIndex);
            MaxMin1D (array, COUNT, &maxyVal, &maxyIndex, &minyVal,
&minyIndex);
                SetAxisScalingMode (panelHandle, PANEL_GRAPH,
VAL_BOTTOM_AXIS,
                VAL_MANUAL, minxVal, maxxVal);
                SetAxisScalingMode (panelHandle, PANEL_GRAPH,
VAL_LEFT_AXIS,
                VAL_MANUAL, minyVal, maxyVal);
            if(i==1){
                wave[0] = a;
                array[0] = b;
            }
            PlotLine(panelHandle, PANEL_GRAPH, wave[d], array[d], wave[i],
array[i], VAL_CYAN);
            //Delay(.02);
        }
        /*----- rand() 2nd -----*/

        GetCtrlVal (panelHandle, PANEL_BINARYSWITCH, &seeding);
        GetCtrlVal (panelHandle, PANEL_COLORNUM, &seed);
            if (seeding == 1) {
                //srand(seed);
                SetRandomSeed(seed);
            }
            x = COUNT/2;
        y = x;
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_6, -1, 1);

        for (i=1; i<COUNT; i++){
            ProcessSystemEvents;
            GetCtrlVal (panelHandle, PANEL_STOPBUTTON,
&stop);

            if (stop == 1){
                ResetTimer (panelHandle, PANEL_TIMER);
                SetCtrlAttribute (panelHandle,
PANEL_TIMER, ATTR_ENABLED, 0);
                break;
            }
        }

```

```

        a = ((Random(0,1)) - 0.5) * 100;
        b = ((Random(0,1)) - 0.5) * 100;
            //a = ((double)rand() - 16384) * SQRT_2;
        //b = ((double)rand() - 16384) * SQRT_2;
        x = x+a;
        y = y+b;
        wave1[i]=x;
        array1[i]=y;
    }
    for (i=1;i<COUNT;i++){
        d = i-1;
        MaxMin1D (wave1, COUNT, &maxxVal2, &maxxIndex2,
&minxVal2, &minxIndex2);
        MaxMin1D (array1, COUNT, &maxyVal2, &maxyIndex2,
&minyVal2, &minyIndex2);
        SetAxisScalingMode (panelHandle,
PANEL_GRAPH_6, VAL_BOTTOM_XAXIS,
        VAL_MANUAL, minxVal2, maxxVal2);
        SetAxisScalingMode (panelHandle, PANEL_GRAPH_6,
VAL_LEFT_YAXIS,
        VAL_MANUAL, minyVal2, maxyVal2);
        if(i==1){
            wave1[0] = a;
            array1[0] = b;
        }
        PlotLine(panelHandle, PANEL_GRAPH_6, wave1[d], array1[d],
wave1[i], array1[i], VAL_CYAN);
        //Delay(.02);
    }
    /* ----- HISTOGRAMS -----*/

    // Reads 2052 bytes for the histogram and normalizing factor of
the raw data

    ClientTCPRead(g_hconversation, &histo, 2048, 5000);
    ClientTCPRead(g_hconversation, &received, 4, 5000);
    //shake = received;
    for (j=0; j<512;j++){
        raw[j] = my_ntohl(histo[j]);
    }
    for (j=0; j<512;j++){
        //d = j-1;
        raw_new[j] = (1.0*raw[j])/(received*1.0) ;
    }
    ArrayToFile ("jem6.txt", histo, VAL_UNSIGNED_INTEGER,
512, 1,

```

```

        VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
        10, VAL_ASCII, VAL_TRUNCATE);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_5, -1, 1);
        PlotY(panelHandle, PANEL_GRAPH_5, raw_new,
512, VAL_DOUBLE, VAL_CONNECTED_POINTS,
        VAL_NO_POINT, VAL_THIN_LINE, 1, VAL_DK_GREEN);
        // Reads 1028 bytes for the histogram and normalizing factor of the whitened
data
        ClientTCPRead(g_hconversation, &histo1, 1024, 1000);
        ClientTCPRead(g_hconversation, &received, 4, 1000);
        shake1 = my_ntohl(received);
        //SetCtrlVal(panelHandle, PANEL_COLORNUM_4, shake1);
        for (j=0; j<256;j++){
            whiten[j] = my_ntohl(histo1[j]);
        }
        for (j=0; j<256;j++){
            //d = j-1;
            whiten_new[j] = (1.0*whiten[j])/(shake1*1.0) ;
        }
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, 1);
        PlotY(panelHandle, PANEL_GRAPH_2, whiten_new,
256, VAL_DOUBLE, VAL_CONNECTED_POINTS,
        VAL_NO_POINT, VAL_THIN_LINE, 1, VAL_DK_GREEN);
        ArrayToFile ("jem7.txt", whiten_new, VAL_DOUBLE, 256, 1,
        VAL_GROUPS_TOGETHER, VAL_GROUPS_AS_COLUMNS,
VAL_SEP_BY_TAB,
        10, VAL_ASCII, VAL_TRUNCATE);
    } //infinite loop
    break;
        case TCP_DISCONNECT:
        MessagePopup ("TCP Client", "Connection Terminated");
        g_connected = 0;
        MainPanelCB (0, EVENT_CLOSE, 0, 0, 0);
        break;
    }
Done:
    if (g_connected) {
        DisconnectFromTCPServer (g_hconversation);
        MessagePopup ("TCP Client", "Connection Terminated");
    }
    return 0;
}

int CVICALLBACK RandTimeCB(int panel, int control, int event, void
*callbackData,

```



```

        int eventData1, int eventData2){
if (event == EVENT_TIMER_TICK){
    SetCtrlAttribute (panelHandle, PANEL_TIMER, ATTR_ENABLED, 0);
    }
    return 0;
}
int CVICALLBACK PseudoCB(int panel, int control, int event, void *callbackData,
    int eventData1, int eventData2){

    int i, j, d, seeding, seeding2, badbmp;
    double x, y, t, a, b, r[max+1], e, f, g;
    double norm = 2147483648;
    if (event == EVENT_COMMIT){
        DisplayImageFile (panelHandle, PANEL_PICTURE,
"randbitmap_true.bmp");
        DisplayImageFile (panelHandle, PANEL_PICTURE_2,
"randbitmap_computer.bmp");
        if (FileSelectPopup ("", "*.txt", "*.txt", "Name of File to Read",
VAL_OK_BUTTON, 0, 0, 1, 0, file_name) > 0){
            //GetCtrlVal (panelHandle, PANEL_InputType, &fileType);

            FileToArray (file_name, temp_history, VAL_DOUBLE, 200, 1,
                VAL_GROUPS_TOGETHER,
VAL_GROUPS_AS_COLUMNS, VAL_ASCII);
            DeleteGraphPlot (panelHandle, PANEL_GRAPH_3, -1, 1);
            //PlotY(panelHandle, PANEL_GRAPH_3, temp_history,
256, VAL_DOUBLE, VAL_CONNECTED_POINTS,
// VAL_NO_POINT, VAL_THIN_LINE, 1, VAL_BLUE);
            x = COUNT/2;
            y = x;

            for (i=1; i<COUNT2; i++){
                ProcessSystemEvents ();
                GetCtrlVal (panelHandle, PANEL_STOPBUTTON,
&stop);
                if (stop == 1){
                    ResetTimer (panelHandle, PANEL_TIMER);
                    SetCtrlAttribute (panelHandle,
PANEL_TIMER, ATTR_ENABLED, 0);
                    break;
                }
                if (i < 100){
                    wave[i] = (double)(temp_history[i]);
                }
            }
            if (i > 100){
                j = i - 100;
                array[j] = (double)(temp_history[i]);
            }
        }
    }
}

```

```

    }
}
for (i=1;i<COUNT;i++){
    x = x + (wave[i] - 1638483247) * SQRT_2;
    y = y + (array[i] - 1638483247) * SQRT_2;
    wave1[i] = x/10000000;
    array1[i] = y/10000000;
}
for (i=1;i<COUNT;i++){
    d = i-1;
    MaxMin1D (wave1, COUNT, &maxxVal, &maxxIndex, &minxVal,
&minxIndex);
    MaxMin1D (array1, COUNT, &maxyVal, &maxyIndex, &minyVal,
&minyIndex);
        SetAxisScalingMode (panelHandle,
PANEL_GRAPH_3, VAL_BOTTOM_XAXIS,
        VAL_MANUAL, minxVal, maxxVal);
        SetAxisScalingMode (panelHandle, PANEL_GRAPH_3,
VAL_LEFT_YAXIS,
        VAL_MANUAL, minyVal, maxyVal);
        PlotLine(panelHandle, PANEL_GRAPH_3, wave1[d], array1[d],
wave1[i], array1[i], VAL_MAGENTA);
        Delay(.02);
    }
}
if (FileSelectPopup ("", "*.txt", "*.txt", "Name of File to Read",
    VAL_OK_BUTTON, 0, 0, 1, 0, file_name) > 0){
    //GetCtrlVal (panelHandle, PANEL_InputType, &fileType);
    FileToArray (file_name, temp_history, VAL_DOUBLE, 200, 1,
        VAL_GROUPS_TOGETHER,
VAL_GROUPS_AS_COLUMNS, VAL_ASCII);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_4, -1, 1);
    x = COUNT/2;
    y = x;
    for (i=1;i<COUNT2;i++){
        ProcessSystemEvents ();
        GetCtrlVal (panelHandle, PANEL_STOPBUTTON, &stop);
    if (stop == 1){
        ResetTimer (panelHandle, PANEL_TIMER);
        SetCtrlAttribute (panelHandle, PANEL_TIMER, ATTR_ENABLED,
0);
                break;
            }
            if (i < 100){
                wave[i] = (double)(temp_history[i]);
            }
    }
}

```

```

        if (i > 100){
            j = i - 100;
            array[j] = (double)(temp_history[i]);
        }
    }
    for (i=1;i<COUNT;i++){
        x = x + (wave[i] - 1638483247) * SQRT_2;
        y = y + (array[i] - 1638483247) * SQRT_2;
        wave1[i] = x/10000000;
        array1[i] = y/10000000;
    }
    for (i=1;i<COUNT;i++){
        d = i-1;
        MaxMin1D (wave1, COUNT, &maxxVal, &maxxIndex, &minxVal,
&minxIndex);
        MaxMin1D (array1, COUNT, &maxyVal, &maxyIndex, &minyVal,
&minyIndex);
        SetAxisScalingMode (panelHandle,
PANEL_GRAPH_4, VAL_BOTTOM_XAXIS,
        VAL_MANUAL, minxVal, maxxVal);
        SetAxisScalingMode (panelHandle, PANEL_GRAPH_4,
VAL_LEFT_YAXIS,
        VAL_MANUAL, minyVal, maxyVal);
        PlotLine(panelHandle, PANEL_GRAPH_4, wave1[d], array1[d],
wave1[i], array1[i], VAL_MAGENTA);
        Delay(.02);
    }
}

if (FileSelectPopup ("", "*.txt", "*.txt", "Name of File to Read",
VAL_OK_BUTTON, 0, 0, 1, 0, file_name) > 0){
//GetCtrlVal (panelHandle, PANEL_InputType, &fileType);
FileToArray (file_name, temp_history, VAL_DOUBLE, 512, 1,
        VAL_GROUPS_TOGETHER,
VAL_GROUPS_AS_COLUMNS, VAL_ASCII);
DeleteGraphPlot (panelHandle, PANEL_GRAPH_5, -1, 1);
PlotY(panelHandle, PANEL_GRAPH_5, temp_history,
512,VAL_DOUBLE,VAL_CONNECTED_POINTS,
        VAL_NO_POINT,VAL_THIN_LINE,1,VAL_DK_GREEN);
}

if (FileSelectPopup ("", "*.txt", "*.txt", "Name of File to Read",
VAL_OK_BUTTON, 0, 0, 1, 0, file_name) > 0){
//GetCtrlVal (panelHandle, PANEL_InputType, &fileType);

FileToArray (file_name, temp_history, VAL_DOUBLE, 512, 1,

```

```

        VAL_GROUPS_TOGETHER,
VAL_GROUPS_AS_COLUMNS, VAL_ASCII);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, 1);
        PlotY(panelHandle, PANEL_GRAPH_2, temp_history,
256,VAL_DOUBLE,VAL_CONNECTED_POINTS,
        VAL_NO_POINT,VAL_THIN_LINE,1,VAL_DK_GREEN);
    }
        SetCtrlAttribute (panelHandle, PANEL_TIMER, ATTR_ENABLED,
1);
        SetCtrlAttribute(panelHandle, PANEL_TIMER, ATTR_INTERVAL, 3.0);
        GetCtrlAttribute (panelHandle, PANEL_TIMER, ATTR_ENABLED, &timer);

        /*----- rand() -----*/

        GetCtrlVal (panelHandle, PANEL_BINARYSWITCH,
&seeding);
        GetCtrlVal (panelHandle, PANEL_COLORNUM, &seed);

        if (seeding == 1) {
            //srand(seed);
            SetRandomSeed(seed);
        }
        x = COUNT/2;
        y = x;
        DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, 1);
        for (i=1;i<COUNT;i++){
            ProcessSystemEvents ();
            GetCtrlVal (panelHandle, PANEL_STOPBUTTON,
&stop);
            if (stop == 1){
                ResetTimer (panelHandle, PANEL_TIMER);
                SetCtrlAttribute (panelHandle,
PANEL_TIMER, ATTR_ENABLED, 0);
                break;
            }
            a = ((Random(0,1)) - 0.5) * 120;
            b = ((Random(0,1)) - 0.5) * 120;
            x = x+a;
            y = y+b;
            wave[i] = x;
            array[i] = y;
        }
        for (i=1;i<COUNT;i++){
            d = i-1;
            MaxMin1D (wave, COUNT, &maxxVal, &maxxIndex, &minxVal, &minxIndex);
            MaxMin1D (array, COUNT, &maxyVal, &maxyIndex, &minyVal, &minyIndex);

```

```

        SetAxisScalingMode (panelHandle, PANEL_GRAPH,
VAL_BOTTOM_XAXIS,
        VAL_MANUAL, minxVal, maxxVal);
        SetAxisScalingMode (panelHandle, PANEL_GRAPH,
VAL_LEFT_YAXIS,
        VAL_MANUAL, minyVal, maxyVal);
        if(i==1){
            wave[0] = a;
            array[0] = b;
        }
        PlotLine(panelHandle, PANEL_GRAPH, wave[d], array[d], wave[i],
array[i], VAL_CYAN);
        Delay(.02);
    }
    /*----- rand() 2nd -----*/

    GetCtrlVal (panelHandle, PANEL_BINARYSWITCH, &seeding);
    GetCtrlVal (panelHandle, PANEL_COLORNUM, &seed);

    if (seeding == 1) {
        //srand(seed);
        SetRandomSeed(seed);
    }
    x = COUNT/2;
    y = x;
    DeleteGraphPlot (panelHandle, PANEL_GRAPH_6, -1, 1);

    for (i=1; i<COUNT; i++){
        ProcessSystemEvents;
        GetCtrlVal (panelHandle, PANEL_STOPBUTTON,
&stop);
        if (stop == 1){
            ResetTimer (panelHandle, PANEL_TIMER);
            SetCtrlAttribute (panelHandle,
PANEL_TIMER, ATTR_ENABLED, 0);
            break;
        }
        a = ((Random(0,1)) - 0.5) * 120;
        b = ((Random(0,1)) - 0.5) * 120;
        //a = ((double)rand() - 16384) * SQRT_2;
        //b = ((double)rand() - 16384) * SQRT_2;
        x = x+a;
        y = y+b;
        wave1[i]=x;
        array1[i]=y;
    }

```

```

        for (i=1;i<COUNT;i++){
            d = i-1;
            MaxMin1D (wave1, COUNT, &maxxVal2, &maxxIndex2,
&minxVal2, &minxIndex2);
            MaxMin1D (array1, COUNT, &maxyVal2, &maxyIndex2,
&minyVal2, &minyIndex2);
            SetAxisScalingMode (panelHandle,
PANEL_GRAPH_6, VAL_BOTTOM_XAXIS,
            VAL_MANUAL, minxVal2, maxxVal2);
            SetAxisScalingMode (panelHandle, PANEL_GRAPH_6,
VAL_LEFT_YAXIS,
            VAL_MANUAL, minyVal2, maxyVal2);
            if(i==1){
                wave1[0] = a;
                array1[0] = b;
            }
            PlotLine(panelHandle, PANEL_GRAPH_6, wave1[d], array1[d],
wave1[i], array1[i], VAL_CYAN);
            Delay(.02);
        }
    }
    return 0;
}
//Function to rearrange the bit order and retrieve original
unsigned long int my_ntohl(unsigned long int in){
    unsigned long int nib, out;
    int i;
    out=0;
    for (i=0; i<32; i=i+8){
        nib = (in>>i)&0x000000FF;
        out = ((out<<8)&0xFFFFF00)|nib;
    }
    return out;
}

```

```

int CVICALLBACK ResetCallback(int panel, int control, int event, void
*callbackData,
    int eventData1, int eventData2){
    if (event == EVENT_COMMIT){
        DeleteGraphPlot (panelHandle, PANEL_GRAPH, -1, 1);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_2, -1, 1);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_3, -1, 1);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_4, -1, 1);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_5, -1, 1);
        DeleteGraphPlot (panelHandle, PANEL_GRAPH_6, -1, 1);
    }
}

```

```

    }
    return 0;
}

```

```

int CVICALLBACK QuitCallback (int panel, int control, int event, void
*callbackData,
    int eventData1, int eventData2){

```

```

    switch (event) {
        case EVENT_COMMIT:
            QuitUserInterface (0);
            break;
    }
    return 0;
}

```

```

static void ReportTCPError (void)

```

```

{
    char messageBuffer[1024];

    if (g_TCPError < 0)
    {
        sprintf(messageBuffer,
            "TCP library error message: %s\nSystem error message: %s",
            GetTCPErrorString (g_TCPError), GetTCPSystemErrorString());
        MessagePopup ("Error", messageBuffer);
        g_TCPError = 0;
    }
}

```

```

int CVICALLBACK MainPanelCB (int panel, int event, void *callbackData,
    int eventData1, int eventData2){

```

```

    if (event == EVENT_CLOSE)
        QuitUserInterface (0);
    return 0;
}

```

Glossary

ASE	Amplified Spontaneous Emission
CW	Continuous Wave
DPSS	Diode Pumped Solid State
EDFA	Erbium Doped Fiber Amplifier
FWHM	Full Width Half Maximum
FWM	Four-wave Mixing
GLP	Green Laser Pointer
H _α	The 656 nm transition in atomic hydrogen
KTP	Potassium Titanyl Phosphate
LWPS	Long Wave Pass Filter
MCA	Multiple-Crystal Assembly
MZM	Mach Zehnder Modulator
Nd	Neodymium
NIR	Near Infrared
OI	Optical Isolator
PBSC	Polarization Beam Splitting Cube
PCF	Photonic Crystal Fiber
PPLN	Periodically Poled Lithium Niobate
PRNG	Pseudo Random Number Generator
QKD	Quantum Key Distribution
SHG	Second Harmonic Generation
SPAD	Silicon Photodiode Avalanche Detector
SWPF	Short Wave Pass Filter
TRNG	True Random Number Generator
YOV	Yttrium Orthovanadate

Bibliography

- [1] http://www.wickedlasers.com/lasers/Spyder_III_Pro_Arctic_Series-96-37.html
- [2] "Machine vision toolbox," P. Corke, *IEEE Robotics and Automation Magazine* 12(4), 16 (2005).
- [3] "A green laser pointer hazard," J. Galang, A. Restelli, E. W. Hagley and C. W. Clark, *Technical Note 1668*, National Institute of Standards and Technology (2010)
- [4] "A red light for green laser pointers," J. Galang, A. Restelli, E. W. Hagley and C. W. Clark, *Optics and Photonics News*(2010)
- [5] "The dangerous dark companion of bright green lasers," J. Galang, A. Restelli, E. W. Hagley and C. W. Clark, *SPIE Newsroom*, January 10 2011
- [6] "A green laser pointer hazard," J. Galang, A. Restelli, E. W. Hagley and C. W. Clark, *International Laser Safety Conference Proceedings*, (2011)
- [7] http://www.repairfaq.org/F_email.html .
- [8] *Code of Federal Regulations*, Title 21, Volume 8, Chapter I, Subchapter J, Part 1040, Sec. 1040.10: Laser products (revised as of April 1, 2009).
- [9] *Laser Safety Management*, Kenneth Barat (CRC Press, Boca Raton, FL, 2006), sec. 9.20; "Ocular Radiation Hazards," David H. Sliney, in *Handbook of Optics, Third Edition, Vol. III: Vision and Vision Optics*, ed. Michael Bass, *et al.* (McGraw Hill, New York, 2010).
- [10] "Laser injuries of the eye," A. B. Thatch, *International Ophthalmology Clinics* **39**(2), 13 (1999)
- [11] "Quasi-phase-matched self-doubling of the frequency in an Nd:Mg:LiNbO₃ laser with a regular domain structure", N. V. Kravtsov *et al.*, *Quantum Electron.* **29** (11), 933 (1999)
- [12] "The compact disc as a diffraction grating", H. Kruglyak, *Physics Education* 26(4), 255 (1991); "Teaching optics in North Uganda", L. Rossi, *et al.* *Optics & Photonics News* 21(6), 14 (2010)
- [13] *Diffraction Grating Handbook, Sixth Edition*, Christopher Palmer (Newport Corporation, Rochester, New York, 2005)

[143] “Quantum cryptography: Public key distribution and coin tossing,” Bennett, C. H., Brassard G., IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India. 175-179 (1984)

[15] “High repetition-rate quantum key distribution,” Bienfang, J.C. Restelli, A. Clark, C.W., Mink A. Quantum Communications Realized. Vol. 6780 of SPIE Proceedings (Society of Photo-optical Instrumentations Engineers, C7800-C7800 (2007)

[16] “High speed quantum key distribution system supports one-time pad encryption of real-time video,” Mink, A., Tang, X., Ma, L.J., Nakassis, T., Hershman, B., Bienfang, J.C., Su, D., Boisvert, R., Clark C.W., and Williams, C.J., Proceedings of SPIE 6244, 62440M (2006).

[17] http://www.noao.edu/image_gallery/html/im0600.html

[18] 656.28 notch

[19] “Reference Wavelengths for Strong Lines of Atomic Hydrogen and Deuterium,” Reader, J., Applied Spectroscopy 58, 1469 (2004)

[20] http://picoquant.com/datasheets/photon_counting/PDM_Series.pdf

[21] http://www.perkinelmer.com/CMSResources/Images/44-12462DTS_SPCM%20AQRH.pdf

[22] “Generation of 250 mW narrowband pulsed ultraviolet light by frequency quadrupling of an amplified erbium-doped fiber laser.” O. Kuzucu, F. N. C. Wong, D. E. Zelmon, S. M. Hegde, T. D. Roberts, and P. Battle, Optics Letters. 32, 1290-1292 (2007).

[23] “Measurement of polarisation-dependent gain in EDFAs against input degree of polarisation and gain compression”, Bruyere, F, IET Electronic Letters, 31(5), 401-403 (1995)

[24] “Temperature and wavelength dependent refractive index equations for MgO-doped Congruent and Stoichiometric LiNbO₃,” Gayer, O, Sacks, Z., Galun, e., Arie, E., Applied Physics B 91, 343-348 (2008)

[25] “Broadband quasi-phase-matched second-harmonic generation in MgO-doped periodically poled LiNbO₃ at the communications band”, Yu, N.E., Ro, J., Cha, M., Kurimura, S., Taira, T Optics Letters, 27 (12), 1046-1048 (2002)

[26] “Generation of correlated photons via four-wave mixing in optical fibers,” Wang, L.J., Hong C.K., Friberg, S.R., Journal of Optics B: Quantum and Semiclassical Optics, 3, 346-352 (2001)

- [27] “Efficient generation of correlated photon pairs in a microstructure fiber,” Fan, J., Migdall, A., Wang, L.J., *Optics Letters* 30(24), 3368 (2005).
- [28] “Microstructure-fiber-based source of photonic entanglement”, Fan, J., Migdall, A., Chen, J. Goldschmidt., *IEEE Journal of Selected Topics in Quantum Electronics*. Vol. 15, No. 6, 1724-1732 (2009)